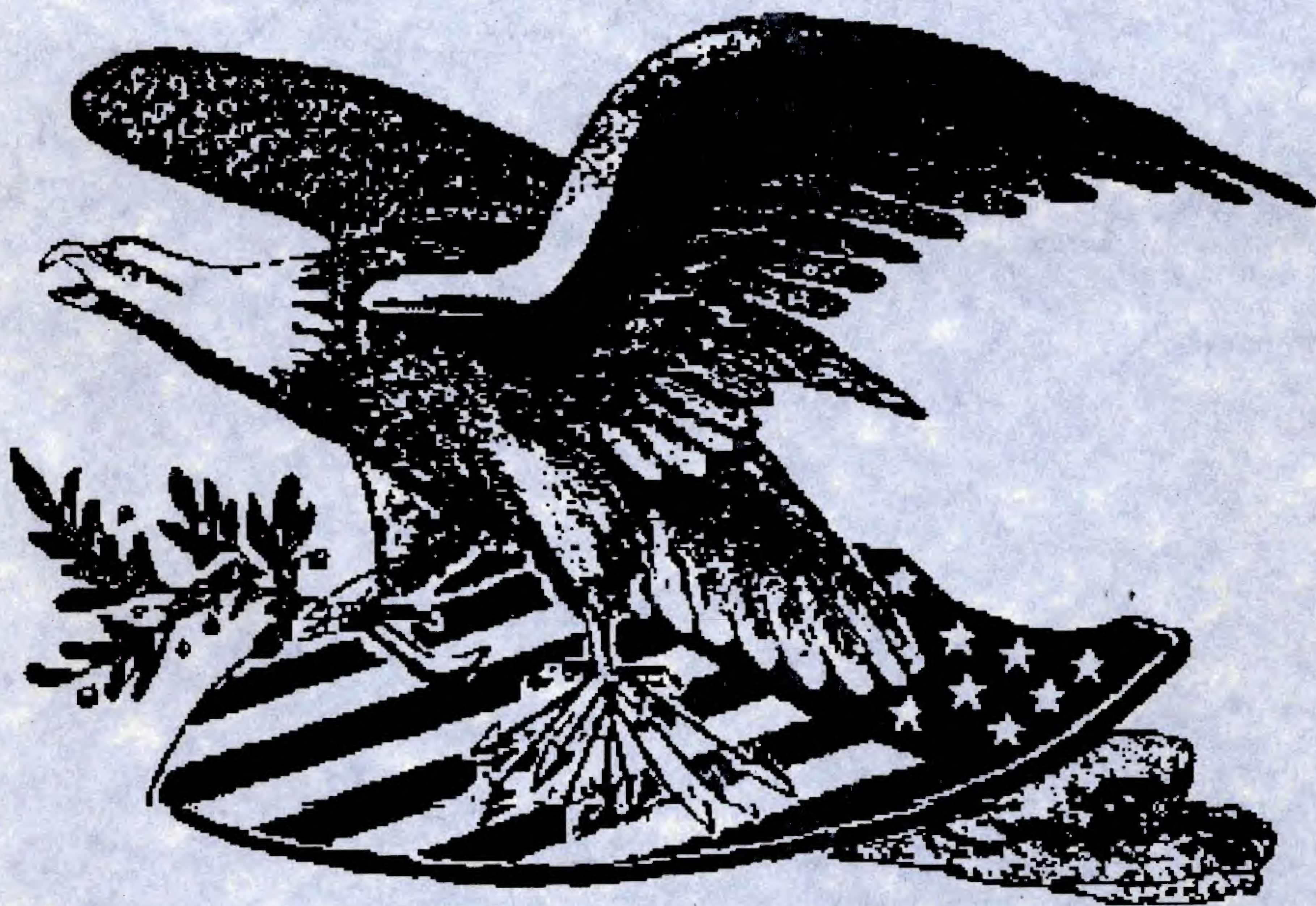


TRSTimes

Volume 5. No. 4. - Jul/Aug 1992 - \$4.00



**HAPPY BIRTHDAY
AMERICA**

LITTLE ORPHAN EIGHTY



It is now almost 14 years since I first joined the 'home computer revolution'. I had messed around with computers before, but they either belonged to an institution or to a friend. The Model I (4K) was the first computer of my very own. As I have chronicled here previously, this gave way to some Color Computers, a Model III, a Commodore 64, a Model 16B, several Model 4's, a couple of Model 100's, and some IBM PC compatibles.

From day one, it seemed that I spent every waking hour learning about the intricacies of my toys; at night, the few hours I slept, I usually dreamt about the computers and what had gone right or wrong with them that day. I was obsessed - I had to learn everything that I possibly could.

In time I learned enough about how they work, and how to program them in various languages, and I am now fortunate enough to make a reasonable living as a teacher, consultant and general computer advisor in the greater Los Angeles area. I consider myself lucky, because I enjoy what I do. I enjoy seeing new users becoming computer buffs, and I enjoy seeing one of my programming students discovering the logic of it all - when he/she gets that 'oh yeah' look, the eyes shining, saying "of course".

The computer industry has changed drastically since I began. Back in 1978 computers were expensive and relatively simple. Today they are cheap and overwhelmingly complex. Consider, I eventually spent close to \$4000 on my Model I, and \$3000 on my Model III. I just helped Roy Beck buy a today's state of the art 486 PC compatible running at 50 mhz, 4 meg of RAM, 256K cache, a 170 meg hard drive, super VGA color monitor, and all the rest of the toys; the cost was less than \$2000. In 6 to 7 months this will be old technology and the price will be somewhere around \$1200. It is amazing and, as I mentioned, overwhelming for today's new user.

Like the hardware, software has come a long way also. Most of us old-timers started with Scripsit and some kind of mailing list. As time went on, some of us graduated to AllWrite or LeScript, Profile, PFS:file, or Little Brother. In the beginning we had no choices to make, we simply learned the available programs. As other programs of the same genre became available, we chose the one that

seemed to do the best job. Learning a new program was relatively easy because we had already worked with a program of the same type. Back then men were men, women were women, word processors processed words, spreadsheets had rows and columns, as well as some built-in math functions, and databases had records and fields.

Today's software is a mirror of our modern world, where women have tattoos, and men are sensitive and wear earrings. Wordprocessors have become desk-top publishers, complete with spelling checkers and thesauruses (thesauri?), columns, graphics, headers, footers, outlines, numbered lists, macros, and a bunch of other stuff. Spread sheets handle three-dimensional information, has built-in graph systems, do limited database work, and can play games -- yes, I have actually played Black Jack with a late version of SuperCalc. Databases are now a far cry from my mailing list program of yesteryear. They have unlimited records with unlimited fields, built-in word processing features, and field to field math functions.

Games, the reason most of us bought computers in the first place, have changed also. Many are the hours I idled away with a Scott Adams adventure or Zork. But just as radio dramas gave way to their television counterparts, the classic text adventures have been replaced with pure graphic games where words are of no importance -- just point the mouse at a particular object, then click the left button, and your character will interact appropriately with the object (now, if this could just be applied to real life...!)

And, speaking of 'pointing and clicking', the mainstay of any real computer, the Disk Operating System, is being replaced by a graphical user interface called Windows. While Windows is said to replace DOS, it cannot execute without DOS, is much slower than DOS, takes up much more hard disk space than DOS, and has more bugs than DOS. But, because Windows uses pictures (icons) instead of words to make the computer productive, it has become very popular.

I am glad that I started back in the 'stone age'. Had I not, I am not so sure that I would have the courage to get involved at the current pace of technology. I pity today's new user; and yet, I am slightly envious of the excitement and fun he/she will have getting familiar with, and besting this now necessary tool of the nineties.

Before closing this column, I would like to sincerely thank the people whose knowledge and hard work helped make this issue possible. A big TRSTimes thank you goes out to Roy Beck, Gary Shanafelt, Karl Mohr, Bruce McDowell, Allen Jacobs, and Mike Ecker.

TRSTimes magazine

Volume 5. No. 4. - Jul/Aug 1992

PUBLISHER-EDITOR
Lance Wolstrup
CONTRIBUTING EDITORS

Roy Beck

Dr. Allen Jacobs

Dr. Michael W. Ecker

TECHNICAL ASSISTANCE

San Gabriel Tandy Users Group

Valley TRS-80 Users Group

Valley Hackers' TRS-80 Users
Group

TRSTimes magazine is published bi-monthly by TRSTimes Publications. 5721 Topanga Canyon Blvd, Suite 4. Woodland Hills, CA. 91364. (818) 716-7154.

Publication months are January, March, May, July, September and November.

Entire contents [c] copyright 1992 by TRSTimes publications.

No part of this publication may be reprinted or reproduced by any means without the prior written permission from the publishers.

All programs are published for personal use only. All rights reserved.

1992 subscription rates (6 issues):
UNITED STATES & CANADA:
\$20.00 (U.S. currency)

EUROPE, CENTRAL & SOUTH
AMERICA: \$24.00 for surface
mail or \$31.00 for air mail.

(U.S. currency only)

ASIA, AUSTRALIA & NEW
ZEALAND: \$26.00 for surface
maile or \$34.00 for air mail.

(U.S. currency only)

Article submissions from our readers are welcomed and encouraged.

Anything pertaining to the TRS-80 will be evaluated for possible publication. Please send hardcopy and, if at all possible, a disk with the material saved in ASCII format. Any disk format is acceptable, but please note on label which format is used.

LITTLE ORPHAN EIGHTY 2
Editorial

THE MAIL ROOM 4
Reader mail

HUNTING FOR BURIED TREASURE 5
Lance Wolstrup

HERE'S LOOKING AT: CLAN4 8
Roy T. Beck

BANK SWITCHING FROM MODEL 4 BASIC 10
Gary W. Shanafelt

NEW COMPUTER LANGUAGES 12
TRSTimes vault

SO, WHAT'S NEW 15
Lance Wolstrup

HINTS & TIPS 16
Morh, McDowell, Jacobs

OBITUARY TO A TRS-80 COLUMN 19
Michael W. Ecker, Ph.D.

SALUTE TO THE FLAG 24
Lance Wolstrup

PROGRAMMING TIDBITS 27
Chris Fara



THE MAIL ROOM



NEED ACCOUNTING UPDATE HELP

My family owns and operates three TRS-80, Model 4 computers and we all agree, they are fine machines.

We appreciate your support of this system and we only hope that it will continue. While I have your attention, there is a small detail that we pray you or some of your readers can offer a solution:

Last year our government made some new rules as it relates to our personal and corporate accounting systems. The method of recording FICA nad Medicare now requires a separate amount for each recorded on our W2 forms and they redesigned the W2 form also. Original Payroll software will not break this out and the original software,m W2 Writer will not comply either. We are searching for software updates to solve this detail. So far, we have been completely unsuccessful. Any assistance will be greatly appreciated.

Tandy software reference numbers:

General Ledger	26-1540
Payroll	26-1543
W2 Writer	26-1539

John Bill Hargrove
6036 Cadiz Street
El Paso, TX 79912

Unfortunately, I am not familiar with this accounting software. My guess is that it is written in either Basic or Cobol, so fixing it should not be a major undertaking. How about it, can anyone out there in TRS-80 land be of help to Mr. Hargrove?

Ed.

BOOT5

This letter is for Adam Rubin, if he is tuned in, of if he cannot be reached, for you, Lance, and your readers. I am sure there are others interested in this subject.

BOOT5/CMD is written by Adam Rubin, and published in the Misosys Quarterly V.iv, p.29. In Computer News 80, December '91 issue (4,12), the Open Forum, p.91, Arthur McAninch made the same appeal. I have seen no response in the CN80 since. Art says that Adam Rubin's address on COMPUSERVE is CIS:71320.1052.

I am using LS-DOS 6.3.1 and LDOS 5.3.0 on my

MISOSYS 40 meg. hard drive, with my 4 or 4P (both Gate-Array), whichever I happen to use. I boot to LS-DOS with a boot disk, which is very quick. I boot to LDOS 5.3.0 from LS-DOS 6.3.1 by typing BOOT5 (i) :1.

I would sure like to use my LDOS 5.3.1, but I get the same message that Art McAninch does when I boot to 5.3.1 using BOOT5. The message is "Drive :1 is not a 5.3.0 disk". If I can find no way to make BOOT5/CMD work with 5.3.1, I am about to do what Art did. That is to use SYS0/SYS from 5.3.0 with the other /SYS files from 5.3.1. If necessary, I'll just boot to LDOS 5.3.1 with a 5.3.1 boot disk.

Has anyone been able to find any patches or modify or find modifications which can be used with LDOS 5.3.1? There are many improvements since LDOS 5.3.0.

I have no modem and know of no one who has, with which to look for info. I sure like it as opposed to a separate boot LDOS disk. It takes nearly no time to boot 6.3.1 and only twice as long to BOOT5 to 5.3.x.

HELP - ANYONE Please

Anthony B. Mizzell
P.O. Box 5302
Chesapeake, VA 23324

Anthony, you can relax. Your problem has been solved. The Jan/Feb 1992 issue of TRSTimes featured an article by Roy Beck called 'BOOT YOUR HARD DRIVE IN LS-DOS AND LDOS WITHOUT A FLOPPY!' (p.10-11). The third paragraph in the left column of page 11 begins a discussion of BOOT5. The continuation on the right column features eight patches that will do exactly what you want - to boot your 4P from LS-DOS 6.3.1 to LDOS 5.3.1. I am repeating the patches here:

```
PATCH BOOT5/CMD (D01,92 = 6A:F01,92 = 62)
PATCH BOOT5/CMD (D03,5F = 31:F03,5F = 30)
PATCH BOOT5/CMD (D04,2B = 31:F04,2B = 30)
PATCH BOOT5/CMD (D04,D3 = D9:F04,D3 = F6)
PATCH BOOT5/CMD (D04,D9 = 50:F04,D9 = 6D)
PATCH BOOT5/CMD (D04,DC = 51:F04,DC = 6E)
PATCH BOOT5/CMD (D04,E1 = D3:F04,E1 = F0)
PATCH BOOT5/CMD (D04,E4 = 12:F04,E4 = 2F)
```

When you enter these patches, you will not be able to boot with 5.3.0. Incidentally, for my own purposes, I added some code to BOOT5/CMD so it will recognize which version of LDOS is to be booted, and will configure itself accordingly. I would like to share this with you and all other readers via an article and a program listing. However, since Adam Rubin's copyright specifically requests that the program not be altered, I will not publish the added code unless I get Mr. Rubin's permission. A letter written Jan 29, 1992 by Roy Beck to Mr. Rubin, requesting permission remains unanswered. I will follow up with another letter, and we'll take it from there.

Ed.

HUNTING FOR BURIED TREASURE

PEEKING AND POKING MODEL 4 SUPER SCROLL-PROTECT BASIC PROGRAMS

By Lance Wolstrup

What seems like an eon ago, I made a deal with 80 Micro for a continuing series on Model 4 Basic programming tricks. They bought the proposal, as well as the outline for the first six chapters. However, a couple of days after I received the acceptance letter, a young man (who shall forever remain nameless) from the editorial staff at 80 Pine Street in Peterborough called me on the phone, telling me that a mistake had been made - the acceptance should really had been a rejection!

Wanting to know why my stuff was being turned down, I pressed the guy to the point where he finally told me that the magazine was heading towards 'ALL MS-DOS', thus they were rejecting all new TRS-80 material. That was in mid-October 1987. TRSTimes was conceived right then and there.!

After a little better than 5 weeks of absolute madness, TRSTimes 1.1 was mailed on January 1, 1988. The feature article was a slightly rewritten version of the first chapter of the rejected 80 Micro series, now titled 'HUNTING FOR BURIED TREASURE'.

Among the many things covered by this article was a way to scroll-protect Basic programs, as could be done in Model III Basic. During 1987, this had been a much debated subject in the pages of 80 Micro. Many methods had been discussed and, while they eventually did the job, they all had one thing in common; they were long and clumsy. The TRSTimes article settled the issue once and for all with this solution:

POKE &HB94,PEEK(&HB94) OR n

n is the number of lines to scroll-protect (0-7) from the top of the screen.

For example, to scroll-protect 5 lines:

POKE &HB94,PEEK(&HB94) OR 5

Scroll-protecting a certain amount of lines from the top of the screen has always been available from Assembly Language, using the DOS SuperVisor Call, @VDCTL. While this SVC controls many other uses, function number 7 allows register C to be loaded with the number of lines to protect (0-7). The long and the short of this routine is that it alters bits 0, 1, & 2 of memory location B94H to reflect the number of protected lines. Our Basic program line does this directly.

By now you will have noticed that the maximum number of lines that can be scroll-protected from the top of the screen is 7 (the protect number is stored in 3 bits in B94H). Neither DOS nor Basic is capable of protecting more.....

WRONG!!!

Basic (and assembler) can scroll-protect as many lines from the top of the screen as you wish. However, in order to do so, you must play a trick on DOS itself. While this makes it just a little more complicated than our one-line Basic program from above, don't worry, it really is not all that difficult.

The trick is to make DOS think that the screen has shrunk - that the lines you wish to protect do not exist. Considering that the Model 4 screen begins at F800H and continues for 1920 (780H) memory locations, the way to accomplish this neat trick is to change five DOS references pointing to the beginning CRT address, as well as two DOS references to the size of the screen.

OK, let's get down to specifics. The following memory addresses reference CRTBGN; the address of the first screen position; all should initially be F800H.

address	value	comment
0B95H	00	this is part of the video
0B96H	F8	driver data table
0C06H	00	DE is loaded with this
0C07H	F8	value. Part of cursor home
		routine.
0C24H	00	HL is loaded with this value.
0C25H	F8	Part of backspace cursor
		routine
0CD4H	00	HL is loaded with this value.
0CD5H	F8	Part of scroll routine.
0D75H	00	DE is loaded with this value.
0D76H	F8	Part of VDCTL routine.

Next we will look at the two sets of memory locations that reference CRTSIZ; the size of the screen; both should initially be 780H (1920 in decimal).

address	value	comment
0CD7H	80	DE is loaded with this value. Part of scroll routine.
0CD8H	07	
0DA8H	80	BC is loaded with this value. Part of the routine to move video ram.
0DA9H	07	

Now that we know the specific memory locations involved, we can discuss the logic of what we need to do.

- The Model 4 screen is 80 chracters wide.
- The normal Model 4 screen begins at F800H. The normal screen size is 1920 (780H).
- To scroll protect R number of lines, add R*80 to F800H and store this new address in the five CRTBGN addresses shown above. Next, subtract R*80 from 780H and store this value in the CRTSIZ addresses shown above.

To do this in Basic, let's assume that the number of lines to scroll protect is stored in variable R. We will pass this value to a subroutine that first computes the new value of CRTBGN and CRTSIZ, then breaks the values into MSB and LSB, which are then POKEd to the appropriate addresses.

```
61 CB = &HF800 + R*80:
IF CD < 0 THEN CB = CB + 65536
'compute new CRTBGN address
```

```
62 CBH = INT(CB/256):
CBL = CB - CBH*256
'store msb in CBH and lsb in CBL
```

```
63 POKE &HB95,CBL:
POKE &HB96,CBH:
POKE &HC06,CBL:
POKE &HC07,CBH:
POKE &HC24,CBL:
POKE &HC25,CBH:
POKE &HCD4,CBL:
POKE &HCD5,CBH:
POKE &HD75,CBL:
POKE &HD76,CBH
'store new CRTBGN address at each DOS reference
```

```
64 CS = &H780 - R*80:
CSH = INT(CS/256):
CSL = CS - CSH*256
'compute new CRTSIZ value
store msb in CSH and lsb in CSL
```

```
65 POKE &HCD7,CSL:
POKE &HCD8,CSH:
POKE &HDA8,CSL:
POKE &HDA9,CSH
'store new CRTSIZ value at each DOS reference
```

```
66 RETURN
```

To demonstrate the power of this subroutine, I am presenting a short program, which I appropriately call SCROLDem/BAS. The listing is found on the next page. Simply type it in and then RUN it from Basic.



Line 10 jumps over the scroll subroutine to the start of the program in line 100.

Do note that line 60 has been added to the scroll subroutine. It makes sure that any value larger than 23 is converted to 0. The reason - 23 is the largest number of lines that anyone needs to scroll protect!

Lines 100-130 scroll protects 10 lines and then proceed to display lines after lines of upper-case A's. This continues until a key is pressed.

Lines 140-160 scroll protects 2 lines and then proceed to display lines after lines of upper-case B's. This continues until a key is pressed.

Lines 170-200 scroll protects 15 lines and then proceed to display lines after lines of CHR(191)'s. This continues until a key is pressed.

Lines 210-240 scroll protects 22 lines and then proceed to display lines after lines of upper-case C's. This continues until a key is pressed.

Line 250 removes all scroll protection.

Line 260 fills the screen with chr\$(191). Notice how ragged the screen looks while filling up with chr\$(191) -- and how slow it is.....

Line 270 waits for a key press.

Line 280 loops to scroll protect from 23 lines all the way down to 0 lines. In each instance, the unprotected portion of the screen is cleared with CLS, causing a smooth line by line erasure -- almost like a theatre curtain being raised before a performance.

SCROLDEM/BAS

```
1 'scroldem/bas
2 '(c) 1992 by Lance Wolstrup
3 'all rights reserved
4 '
10 GOTO 100
60 IF R > 23 THEN R = 0
61 CB = &HF800 + R*80:
IF CB < 0 THEN CB = CB + 65536!
62 CBH = INT(CB/256):
CBL = CB-CBH*256
63 POKE &HB95,CBL:
POKE &HB96,CBH:
POKE &HC06,CBL:
POKE &HC07,CBH:
POKE &HC24,CBL:
POKE &HC25,CBH:
POKE &HCD4,CBL:
POKE &HCD5,CBH:
POKE &HD75,CBL:
POKE &HD76,CBH
64 CS = &H780-R*80:
CSH = INT(CS/256):
CSL = CS-CSH*256
65 POKE &HCD7,CSL:
POKE &HCD8,CSH:
POKE &HDA8,CSL:
POKE &HDA9,CSH
66 RETURN
100 CLS:PRINT"Scroll protecting 10 lines...."
110 FOR X = 1 TO 10:
PRINT STRING$(80,191);:
NEXT
```

```
120 R = 10:
GOSUB 60
130 IF INKEY$ = "" THEN PRINT STRING$(80,65);:
GOTO 130
140 PRINT@0,"Now scroll protecting 2 lines"
150 R = 2:
GOSUB 60
155 PRINT@(23,0),"";
160 IF INKEY$ = "" THEN PRINT STRING$(80,66);:
GOTO 160
170 PRINT@0,"Scroll protecting 15 lines...."
180 R = 15:
GOSUB 60
190 PRINT@(23,0),"";
200 IF INKEY$ = "" THEN PRINT STRING$(80,191);:
GOTO 200
210 PRINT@0,"Scroll protecting 22 lines...."
220 R = 22:
GOSUB 60
230 PRINT@(23,0),"";
240 IF INKEY$ = "" THEN PRINT STRING$(40,67):
GOTO 240
250 R = 0:
GOSUB 60
260 CLS:
FOR X = 0 TO 23:
PRINT STRING$(80,191);:
NEXT
270 IF INKEY$ = "" THEN 270
280 FOR R = 23 TO 0 STEP -1:
GOSUB 60:
CLS:
NEXT
```



HERE IS LOOKING AT: CLAN4

Review by Roy T. Beck

How many genealogists are there among us TRS users? That's a rhetorical question, of course, but it touches upon one of the areas where the TRS-80 was previously almost unrepresented, and that is Genealogy.

Being a genealogist, among other things, I have long felt the lack of a good genealogy program for the Mod 4. Long ago (maybe 10 years?), there was a BASIC genealogy program for the Model I, written, I believe, by a man named Richardson. Also, there is a "freeware" program named CLAN, written for the Model III by Arthur C. Hurlburt.

Chris Fara has recently undertaken to fill this gap in the range of available Model 4 programs, and CLAN4 is the result. Thank you, Chris!

CLAN4 is written entirely in machine code, and is rapid in its response. Of course, I ran it on my hard drive, which naturally speeds up file access. While hard drive operation is the way to go for any program which does as much file accessing as a genealogy program (which is really a special form of data base program), CLAN4 will also run happily on a floppy system, just a little slower.

As mentioned above, CLAN4 is the successor to CLAN for the Model III, and Chris has included a conversion program to allow the user to convert existing CLAN files to CLAN4. This is really an essential feature, as reentering all of a genealogist's file data into a new program is a daunting chore.

CLAN4 can actually operate with up to four different sets of data. At sign-on time, the last used data set is proposed, but the user can cycle through the other three data sets, allowing him to open whichever he wants. Of course, even more sets could be accessed, if desired, by installing multiple copies of CLAN4, using diskDISK, for example. Unless one is a professional genealogist, four data sets certainly ought to be sufficient.

Installation was a snap. Just BACKUP the master disk to your floppy or hard disk and you are ready to go. Not having any Model III CLAN data, I was unable to check out the conversion routine. However, Chris' reputation inspires me to believe the conversion should be flawless. The best approach on a hard drive is to assign one diskDISK to CLAN4; on a floppy system, one double sided floppy would be great, because the entire CLAN4 and one set of files would fit on a DSDD floppy.

How many people can CLAN4 accommodate? This is a valid question because any genealogist worth his salt can quickly accumulate a significant number of people

in just one "clan". CLAN4 can handle up to 1792 people and up to 448 surnames. The limit on children is 16 per couple and up to 50 per any one person. This should handle Mormon marriages easily. (One of my own Mormon great grandfathers sired 23 children by two wives). A feature known as "Legend" accommodates up to about 2000 ASCII characters and is available to hold free-form text (notes, references, anecdotes, etc) for each person.

A fully expanded set of CLAN4 data files will occupy up to 150K of floppy disk space, somewhat more on a hard drive due to the typically larger granule sizes used on hard drives.

An important feature of CLAN4 is the HELP key, F1, available at all times with useful guidance. I found this assistance to be essential until I had learned the commands. A very useful feature, in my opinion. Chris also provides a pocket folder type of instruction sheet, but I found myself using the F1 key in preference, as it is "context sensitive", changing as required to suit whatever situation currently exists.

CLAN4 differs from manual genealogy in that everything is focussed on an entry of an individual "Personal Record". Once the individual's personal record is in the file, then the genealogist may proceed to establish linkages, that is, marriage to another person in the data base, or parentage with others in the data base. Once the user accustoms himself to mandatory entry of persons before specifying the linkages which relate persons, the program goes along easily.

A peculiar constraint is the requirement that a date of birth must be stated for a person before he can be accepted into the data base. As a result of this requirement, I have a lot of ancestors born in 1800?. The date is required to enter them, and the ? indicates to me that the date is not to be believed. Most programs assign what amounts to a serial number to every person, and do not require anything more than a name; apparently Chris opted for this mandatory date of birth method as a means of uniquely identifying persons with the same name. Again, once you understand what is going on, then the program moves along well. The same type of constraint appears when entering marriages between two people. A date must be specified, with a ? if the date is not accurately known. Fortunately, the marriage need not be entered to establish paternity or maternity. For any given person, it is sufficient to note the name of the father or mother (although that person must have been previously entered into the data base).

Indicating maternity or paternity directly avoids the delicate question of whether or not any two people were married. If you know they were (or are), then you can enter it. If you don't know, then just ignore the question until it is established. The descendancy is still shown correctly, regardless of their marital state.

Stillborn children and infants who died young can be handled with no difficulty. As is customary, all such children are assumed male unless the contrary is known. A surname and date of birth are required. The given name can be simply "stillborn" or "infant" or whatever identifier you select if no name was recorded. The name and date of birth identify the child. The death date goes further to identify the child. If the child is stillborn, naturally the date of birth and death are the same.

Children need not be entered in any particular sequence. The program will sequence them correctly by the dates of birth.

The program does not produce a "family sheet" of the usual type; it produces instead a "Resume", which contains approximately the same information one would expect to find on a family sheet, including spouses, parents, children, etc.

The program produces both pedigree and descendancy charts, which both begin with a specific individual. In the pedigree (here called an "ancestors" chart), the usual display is provided, with the paternal line on the top of the page and the maternal line on the lower half.

The descendants chart lists each successive descendant chronologically, with lateral offsets on the printout to distinguish generations. The user can specify the number of generations to be printed, from 1 to 9. Specifying one generation results in only the children of the individual being listed; two adds grandchildren to the chart, and so on. Placing the Original Immigrant at the top of the chart will produce an impressive chart, with all descendants neatly tabulated.

One significant omission is the lack of a data interchange facility. This feature is widely available in programs written for the IBM family, and greatly facilitates importing or exporting of data to other researchers, either by disk or RS-232.

Documenting your sources of data is extremely important in serious genealogy, and CLAN allows use of the "Legend" area for this purpose. This is a free-form ASCII text file which is associated with each individual in the file.

All of the printable outputs can be "printed" to a disk file, allowing later massaging via your favorite word processor, assuming it will accept a straight ASCII file, which most will do.

Printers can be a source of difficulty in our modern computer world. CLAN provides an option list of four

different "generic" printers when installing it. This is further extended by the fact that you can install special escape sequences unique to your printer. This actually allows the user to choose from an almost unlimited number of printers, including even the HP Lasers. That is nice!

Editing of entries is by means of the arrow keys and the function keys. Both Insert and Typeover modes are available (toggled by the F3 key). F2 is the delete key, deleting the letter it is on in the typeover mode and the letter to its left in the insert mode. The editor is basically a page editor, and works well.

Date entry is easy, using the US convention of mm dd yyyy. Separators can be "/", " ", "-", or " ", all of which work well. The printout will be in the form of: Sept. 23, 1992. Further, years in the 20th century can be input as yy, and will be converted to yyyy automatically. Years from 100 to 3000 can be entered.

Because the assignment of parents to children and spouses to marriages requires identifying that person in the file, a rapid search is necessary. This search (which is case sensitive) will allow you to quickly find the correct person in the Personal Record.

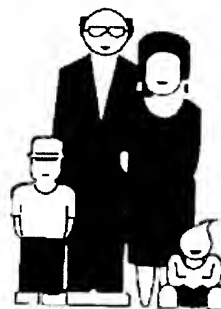
Another nice touch is the inclusion of a screen saver feature. The default is 5 minutes, but the time is adjustable. At the touch of any active key, the screen is restored.

Altogether, this is quite a nice program to have sprung fullblown into our TRS market. Chris is to be congratulated upon a significant piece of work. For you amateur genealogists who do not have an IBM or clone, this is the way to go. With a good market response, Chris can probably add things like data interchange capability. Let's support him.

The cost is \$69.95. It is available from:

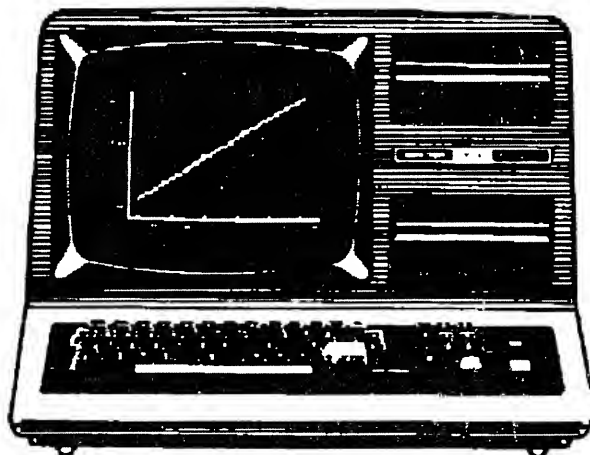
MICRODEX
1212 Sawtelle Ave
Tucson, AZ 85717
(602) 326-3502

The Model III CLAN conversion program is an optional extra for \$29.95.



BANK SWITCHING FROM MODEL 4 BASIC

By Gary W. Shanafelt



One of the limitations of Model 4 BASIC is that you normally don't have very much memory available for your programs. Even more frustrating, if you have a 128k machine, that extra 64k of memory isn't used by BASIC. The Z80 can access only 64k of memory at a time, and it uses the full 128k through a process called "bank switching." The memory in the machine is organized in banks of 32k each, and at command it can switch one bank out and switch another in. The LSDOS operating system includes a supervisor call to accomplish this, @BANK, and a number of programs like Visicalc use this DOS routine to double the memory for your data if you have a 128k machine.

Unfortunately, BASIC isn't one of those programs.

The reason is fairly simple. Since the operating system resides in the lower 32k of memory, @BANK switches the upper 32k (addresses 8000H-FFFFH) with the alternate memory banks in a 128k machine. The routine doing the switching, and the stack, MUST be located below 8000H: otherwise, as soon as a switch is made, they are switched out and the whole system crashes. Well, BASIC stores its

stack and all its variables above 8000H. So, unless you happen to have a special add-on routine Misisys produced a while back, the @EXMEM SVC, you're out of luck.

I was out of luck. @EXMEM is easy enough to get from the Misisys Quarterly's Disk Notes, but I wanted to add some features to a program called MDRAW II and I couldn't very well tell potential users they would have to possess @EXMEM to use them. MDRAW II, by the way, is a drawing program for the Model 4 hi-res board written by Scott McBurney. It requires GBASIC and LSDOS 6.3 but has some nifty features like displaying Dotwriter fonts on the screen in addition to all the usual functions you expect in a drawing program. And of course, for the last year it has been the only program available using the serial mouse drivers developed by David Goben and Matthew Reed. It is available for free on all reputable TRS-80 BBS's (if yours doesn't have it, tell your sysop to get it!)

I have been working over Scott's code; version 4 now includes repositioning the whole screen and zooming to edit individual pixels. What I really wanted MDRAW to do, though, was to use the extra memory on a 128k machine to allow editing of several graphics screens at the same time, so you could move blocks of data between them by just hitting a key. The screen not currently on display would be stored in one of the alternate memory banks. But how could I get to those extra banks of memory from within BASIC?

I obviously found a solution, or I wouldn't be writing this. Rather than trying to switch out the upper bank of regular memory, from 8000H-FFFFH, I decided to try to switch the LOWER one. There is no routine in LSDOS to do this, because this means essentially switching the operating system out from under BASIC. You thus have to address the requisite port directly. As long as you disable the interrupts and don't use any DOS calls, you can then run a machine language routine in BASIC which has access to a full extra 32k of memory -- from 0000H-7FFFH. In the case of MDRAW, my routine switched the lower memory with one of the alternate banks, toggled the memory in the bank with the memory on the hi-res screen, switched the banks back, and finally returned to BASIC. The whole routine took about 50 bytes. As far as the DOS was concerned, nothing had happened because it was switched out with the interrupts disabled while all this was going on. You can now toggle between three hi-res screens so fast in MDRAW that even my wife is mildly impressed.

How do you do all this? First, you've got to incorporate your machine language routine into BASIC, either poking it into high memory or storing it in a string. Then, you call it. Before the call, you need to use the LSDOS BASIC SVC interface to call the @BANK routine (supervisor call 102) to check if the bank you want to switch is available. If the bank is already being used by something else, such as Misisys' Overdrive utility or a Memdisk, you could crash

your system by overlaying those routines with your own data. The SVC call in BASIC for @BANK looks like this:

100 C = 1 ' for bank 1; use C = 2 for bank 2

110 SVC(0) = 102:SVC(3) = 512 + C:
JNK = USR11(VARPTR(SVC(0)))

120 IF (SVC(0) AND 64) = 0 THEN STOP
' The bank is not available: abort routine!

130 ' Call the bank-switching routine here

The bank-switching routine looks like this:

```
C5      PUSH  BC          ; Save registers,
                          ; just in case...

D5      PUSH  DE
E5      PUSH  HL
F5      PUSH  AF

F3      DI                ; Disable interrupts:
                          ; forget this, and
                          ; you're doomed!

3E67    LD      A,67H      ; Select bank 1 to switch
                          ; with low memory
                          ; Use 3E77 -- LD A,77H
```

```
D384    OUT     (84H),A    ; to switch bank 2 with
                          ; low memory
                          ; Send it to port 84H:
                          ; this switches the banks
```

[Your routine goes here. Remember that the free memory is addressed 0000H-7FFFH; put anything higher than that, and you'll corrupt your BASIC program. And you can't make any calls to the operating system. To end, simply reverse the process:]

```
3E87    LD      A,87H      ; 87H is the normal
                          ; configuration of
                          ; memory banks

D384    OUT     (84H),A    ; Send it to port 84H:
                          ; this switches the banks
                          ; back

FB      EI                ; Enable the interrupts
                          ; again

F1      POP     AF         ; Restore the registers
E1      POP     HL
D1      POP     DE
C1      POP     BC

C9      RET                ; Back to BASIC!
```

MISOSYS, Inc.

Aerocomp Hardware is now available from MISOSYS

- Model I DDen Controller (DDC) \$45 + \$6S&H
- Model III/4 FDC board \$45 + \$6S&H
- Model III/4 RS232 board \$45 + \$6S&H
- Model III/4 RS232 Kit \$50 + \$6S&H
- WD1002S-SHD HDC (new) \$75 + \$5S&H
- Aerocomp 5 Meg HD \$250 + S&H
- Aerocomp 20 Meg HD \$400 + S&H
- Aerocomp 40 Meg HD \$500 + S&H
- MM CP/M 2.2 HD drivers \$29.95 + \$3S&H

The Comsoft Group

Model I/III Action Games Special

All five games on a single disk for one low price!
• You get Bounceoids, Crazy Painter, Space Castle, Scarfman,
and The Official Frogger; all five for \$20 + \$3S&H

Sale prices
good
through
Aug
31st.



MISOSYS, Inc.

P. O. Box 239
Sterling, VA 20167-0239
703-450-4181
orders: 800-MISOSYS

The MISOSYS Quarterly subscriptions

Keep up to date with the latest information on MISOSYS products, programs, patches, and articles in a professional magazine format. A subscription to *TMQ* will provide you with information, news, and announcements concerning our entire product line and related machine environments. As a special for new subscribers, we'll provide you with five issues and start you off with issue VI.iv, and send you three past issues at no charge. That's eight issues for the price of four! Subscriptions are: \$25 US; \$30 Canada; \$35 Europe; \$40 Australia

PRO-WAM Pop up Price Plunge

If you are not using PRO-WAM on your 128K Model 4, you're not using your 4! You'll get a pop-up desktop manager with ADDRESS, BRINGUP, CAL, CARD, CALC, PHRASE, and more. Export/Import across windows. PSORT your data files. \$37.48 + \$6S&H

DoubleDuty doubles your 128K Model 4

Now on sale at half price!

DoubleDuty divides your Mod4's memory into three complete and independent partitions. Two operate as they were each their own 64K computer. Get the best task switcher you can buy. Our 2.6 release also works with extended memory. If you thought you needed a second computer, think again. At \$24.98 + \$3S&H, you can't afford to not have DoubleDuty.

NEW COMPUTER LANGUAGES

Humor from the TRSTimes vault

B A B B A G E THE LANGUAGE OF THE FUTURE

There are few things in this business that are more fun than designing a new computer language, and the very latest is Ada - the Department of Defense's new supertoy. Ada, as you know, has replaced outmoded and obsolete languages such as COBOL and FORTRAN.

The problem is that this cycle takes 20 to 30 years and doesn't start until we're really convinced present languages are no good. We can short-circuit this process by starting on Ada's replacement right now. Then, by the time we decide Ada is obsolete, its replacement will be ready.

The new generation of language designers has taken to naming its brainchildren after real people rather than resorting to the usual acronyms. Pascal is named after the first person to build a calculating machine and Ada is named after the first computer programmer. As our namesake, we chose Charles Babbage, who died in poverty while trying to finish building the first computer. The new language is thus named after the first systems designer to go over budget and behind schedule.

Babbage is based on language elements that were discovered after the design of Ada was completed. For instance, C.A.R. Hoare, in his 1980 ACM Touring Award lecture, told of two ways of constructing a software design: "One way is to make it so simple that there are obviously no deficiencies and the other way is to make it so complicated that there are no obvious deficiencies."

The designers of Babbage have chosen a third alternative - a language that has only obvious deficiencies. Babbage programs are so unreliable that maintenance can begin before system integration is completed. This guarantees a steady increase in the DP job market.

Like Pascal, Ada uses "strong typing" to avoid errors caused by mixing data types. The designers of Babbage advocate "good typing" to avoid errors caused by misspelling the words in your program. Later versions of Babbage will also allow "touch typing" which will fill a long-felt need.

A hotly contested issue among language designers is the method for passing parameters to subfunctions. Some advocate "call by name," others prefer "call by value."

Babbage uses a new method - "call by telephone." This is especially effective for long-distance parameter passing.

Ada stresses the concept of software portability. Babbage encourages hardware portability. After all, what good is a computer if you can't take it with you?

It's a good sign if your language is sponsored by the government. COBOL had government backing, and Ada is being funded by the Department of Defense. After much negotiation, the Department of Sanitation has agreed to sponsor Babbage.

No subsets of Ada are allowed. Babbage is just the opposite. None of Babbage is defined except its extensibility - each user must define his own version. To end the debate of large languages versus small, Babbage allows each user to make the language any size he wants. Babbage is the ideal language for the "me" generation. The examples that follow will give some idea of what Babbage looks like.

Structured languages banned GOTOs multiway conditional branches by replacing them with the simpler IF-THEN-ELSE structure. Babbage has a number of new conditional statements that act like termites in the structure of your program:

WHAT IF - Used in simulation languages. Branches before evaluating test conditions.

OR ELSE - Conditional threat, as in: "Add these two numbers OR ELSE!"

WHY NOT? - Executes the code that follows in a devil-may-care fashion.

WHO ELSE? - Used for polling during I/O operations.

ELSEWHERE - This is where your program really is when you think it's here.

GOING GOING GONE - For writing unstructured programs. Takes a random branch to another part of your program. Does the work of 10 GOTOs.

For years, programming languages have used "FOR", "DO UNTIL", "DO WHILE", etc. to mean "LOOP". Continuing with this trend, Babbage offers the following loop statements:

DON'T DO WHILE NOT - This loop is not executed if the test condition is not false (or if it's Friday afternoon).

DIDN'T DO - The loop executes once and hides all traces.

CAN'T DO - The loop is pooped.

WON'T DO - The CPU halts because it doesn't like the code inside the loop. Execution can be resumed by typing "MAY I" at the console.

MIGHT DO - Depends on how the CPU is feeling. Executed if the CPU is "UP", not executed if the CPU is "DOWN" or if it's feelings have been hurt.

DO UNTIL OTHERS - Used to write the main loop for timesharing systems so that they will antagonize the users in a uniform manner.

DO-WAH - Used to write timing loops for computer-generated music (Rag Timing).

Every self-respecting structured language has a case statement to implement multiway branching. ALGOL offers an indexed case statement and Pascal has a labeled case statement. Not much of a choice. Babbage offers a variety of case statements:

The JUST-IN-CASE Statement - For handling afterthoughts and fudge factors. Allows you to multiply by zero to correct for accidentally dividing by zero.

The BRIEF CASE Statement - To encourage portable software.

The OPEN-AND-SHUT CASE Statement - No proof of correctness is necessary with this one.

The IN-ANY-CASE Statement - This one always works.

The HOPELESS CASE Statement - This one never works.

The BASKET CASE Statement - A really hopeless case.

The Babbage Language Design Group is continuously evaluating new features that will keep its users from reaching any level of effectiveness. For instance, Babbage's designers are now considering the **ALMOST EQUALS SIGN**, used for comparing two floating point numbers. This new feature "takes the worry out of being close."

No language, no matter how bad, can stand on its own. We need a really state-of-the-art operating system to support Babbage. After trying several commercial systems, we decided to write a "virtual" operating system. Everybody has a virtual memory operating system so we decided to try something a little different. Our new operating system is called the **Virtual Time Operating System (VTOS)**. While virtual memory systems make the computer's memory the virtual resource, VTOS does the same thing with CPU processing time.

The result is that the computer can run an unlimited number of jobs at the same time. Like the virtual memory system, which actually keeps part of the memory on disk,

VTOS has to play tricks to achieve its goals. Although all of your jobs seem to be running right now, some of them are actually running next week.

As you can see, Babbage is still in its infancy. The Babbage Language Design Group is seeking suggestions for this powerful new language and as the sole member of this group (all applications for membership will be accepted), I call on the data processing community for help in making this dream a reality.

NEW LANGUAGES COMPETE WITH APL

APL, BASIC, FORTRAN, COBOL,... these programming languages are well known and (more or less) well loved throughout the computer industry. There are numerous other languages, however, that are less well known yet still have ardent devotees. In fact, these little-known languages generally have the most fanatic admirers. For those who wish to know more about these obscure languages -- and why they are obscure -- I present the following catalog.

SIMPLE

SIMPLE is an acronym for Sheer Idiot's Monopurpose Programming Linguistic Environment. This language, developed at Hanover College for Technological Misfits, was designed to make it impossible to write code with errors in it. The statements are, therefore, confined to **BEGIN**, **END**, and **STOP**. No matter how you arrange the statements, you can't make a syntax error.

Programs written in **SIMPLE** do nothing useful. They thus achieve the results of programs written in other languages without the tedious, frustrating process of testing and debugging.

SLOBOL

SLOBOL is best known for the speed, or lack of it, of its compiler. Although many compilers allow you to take a coffee break while they compile, **SLOBOL** compilers allow you to travel to Bolivia to pick the coffee. Forty-three programmers are known to have died of boredom sitting at their terminals while waiting for a **SLOBOL** program to compile.

VALGOL

From its modest beginnings in Southern California's San Fernando Valley, **VALGOL** is enjoying a dramatic surge of popularity across the industry.

VALGOL commands include **REALLY**, **LIKE**, **WELL**, and **Y*KNOW**. Variables are assigned with the **= LIKE** and

=TOTALLY operators. Other operators include the California Booleans, FERSURE and NOWAY. Repetitions of code are handled in FOR - SURE loops. Here is a sample VALGOL program:

```
LIKE Y*KNOW (I MEAN) START
IF PIZZA =LIKE BITCHEN AND B =LIKE TUBULAR
AND C =LIKE GRODY**MAX THEN
    FOR I =LIKE 1 TO OH MAYBE 100
        DO WAH - (DITTY**2)
        BARF(I) =TOTALLY GROSS(OUT)
    SURE
LIKE BAG THIS PROBLEM
REALLY
LIKE TOTALLY(Y*KNOW)
```

VALGOL is characterized by its unfriendly error messages. For example, when the user makes a syntax error, the interpreter displays the message:

GAG ME WITH A SPOON!

LAIDBACK

Historically, VALGOL is a derivative of LAIDBACK, which was developed at the (now defunct) Marin County Center for T'ai Chi, Mellowness, and Computer Programming, as an alternative to the intense atmosphere in nearby Silicon Valley.

The center was ideal for programmers who liked to soak in hot tubs while they worked. Unfortunately, few programmers could survive there for long, since the center outlawed pizza and RC Cola in favor of bean curd and Perrier.

Many mourn the demise of LAIDBACK because of its reputation as a gentle and nonthreatening language. For example, LAIDBACK responded to syntax errors with the message:

SORRY MAN, I CAN'T DEAL BEHIND THAT.

SARTRE

Named after the late existential philosopher, SARTRE is an extremely unstructured language. Statements in SARTRE have no purpose; they just are. Thus SARTRE programs are left to define their own functions. SARTRE programmers tend to be boring and depressed and are no fun at parties.

FIFTH

FIFTH is a precision mathematical language in which the data types refer to quantity. The data types range from CC, OUNCE, SHOT, and JIGGER to FIFTH (hence the name of the language), LITER, MAGNUM, and BLOTTO. Commands refer to ingredients such as CHABLIS, CHARDONNAY, CABERNET, GIN, VERMOUTH, VODKA, SCOTCH, BOURBON, CANADIAN, and WHATEVER SAROUND.

The many versions of the FIFTH language reflect the sophistication and financial status of its users. Commands in the ELITE dialect include VSOP, LAFITE, and WAITER'S(RECOMMENDATION). The GUTTER dialect instead has commands for THUNDERBIRD, RIPPLE, and HOUSE(RED). The GUTTER dialect is a particular favorite of frustrated FORTH programmers who end up using this language.

C-

This language was named for the grade received by its creator when he submitted it as a project in a graduate programming class. C- is best described as a "low-level" programming language. In general, the language requires more C- statements than machine-code instructions to execute a given task. In this respect it is very similar to COBOL.

DOGO

Developed at the Massachusetts Institute of Obedience Training, DOGO heralds a new era of computer-literate pets. DOGO commands include SIT, STAY, HEEL, and ROLL OVER. An innovative feature of DOGO is "puppy graphics", a small cocker spaniel that occasionally leaves deposits as he travels across the screen.

3 Biggest Software Lies

- The program's fully tested and bugfree.
- We're working on the documentation.
- Of course we can modify it.

3 Biggest Computer Room Lies

- As long as you remember to 'SAVE' your input, you'll never lose any files.
- We run the stuff through as fast as it comes in the door.
- The new machine's on order.

3 Biggest Large Company Lies

- We have an entrepreneurial spirit here.
- People are our greatest resource.
- We say 'let the marketplace decide'.

3 Biggest Small Company Lies

- We have an entrepreneurial spirit here.
- The boss is just one of the guys.
- Staying small is a conscious decision.

3 Biggest Marketing Lies

- Immediate delivery?...No problem.
- We treat every customer as if they were our most important.
- We're going out to lunch to talk business.

SO, WHAT'S NEW?

The LDOS & LS-DOS Reference Manual

a review by Lance Wolstrup

It was a little better than a year ago that Roy Soltoff of Misosys, Inc. brought us the long awaited upgrade to our Model 4 DOS, now becoming LS-DOS 6.3.1. A short time thereafter, to make Model I/III and 4 machines as compatible as possible, Mr. Soltoff also brought out upgrades to Model I and III LDOS, each becoming version 5.3.1.

While these, without a doubt, are the finest DOSes our machines have ever worked with, a problem crept up. Actually, this problem began when Tandy upgraded Model 4 TRSDOS from 6.0 to 6.1: we did not get a new manual -- we simply got a couple of pages containing upgrade information that was to be inserted into our current manual.

TRSDOS 6.1 evolved to 6.1.1, and then to 6.1.2, each with new pages, explaining the changes, to be inserted into the manual.

TRSDOS 6.1.2 became TRSDOS 6.2.0, and then, finally, 6.2.1. Again, more pages to add to the manual. Oh, I know, some of you lucky people who bought your Model 4 rather late, got an actual TRSDOS 6.2 manual from Radio Shack, but the rest of us 'early-birds' had to settle for upgrade sheets.

Primarily because TRSDOS 6.2.1 could not accept dates past December 31, 1987, the DOS was once again upgraded. This time it was a major revision, and TRSDOS 6.2.1 became LS-DOS 6.3.0. Again, we received update-sheets rather than a new manual. At this time my manual was becoming somewhat unruly.

As mentioned above, the latest version of DOS for Model 4 is 6.3.1. With this version we received a 38-page, 5 1/2 x 8 1/2 booklet, containing the update documentation.

Model III LDOS follows the same path; the last version with a manual included was LDOS 5.1.4. The upgrade to 5.3.0 brought insert-sheets, while the upgrade to 5.3.1 brought a booklet similar the LS-DOS 6.3.1 book.

The last manual for Model I LDOS came with version 5.1.4. There was never a version 5.3.0, rather LDOS 5.1.4 was upgraded directly to 5.3.1. This upgrade came with an upgrade booklet identical to the Model III booklet.

By now it should be plain for everybody to see what the problem is: Outdated manuals with insert-sheets of different sizes, some probably missing.

Good news!! The problem is now solved. Roy Soltoff has just released a combination Model I/III & 4 manual. It is sized at 5 1/2 x 8 1/2, and has 339 pages of dynamite information. Yes indeed, it covers Model I/III and 4, all in one. This is especially valuable to someone, like me, who works with all three machines. It is nice to have SOLE and MEMDISK explained in the same manual.

The LDOS & LS-DOS Reference Manual, as it is properly called, begins with a section titled 'Introduction to DOS', which is broken down into subsections called 'Using this manual', 'What is a Disk Operating System (DOS)?', 'Disk organization', 'Files - How information is stored', 'DOS organization and files', 'DOS and Devices', 'Using the DOS files', etc. For a new user, this chapter is invaluable -- a goldmine of information; as a matter of fact, even old hands can pick up some knowledge here.

The next section explains what the thirteen system files do, followed by a short description of each of the utility files, filters and drivers that come with the DOS. Bet you don't know what all of them do -- I was ignorant of a couple! This section also gives the passwords to all DOS files.

System devices *KI, *DO, *PR, *JL, *SI, and *SO are discussed in detail, followed by a tutorial on disk drives and memory usage. Here I have a legitimate question - on page 35, the term HIGH\$ is discussed and we are told that it is pronounced "High dollar"! I always took it for granted that it was pronounced "High String". Am I wrong?

The next 207 pages cover the DOS library commands and utilities. Each is discussed in detail and with plenty of examples. This is the heart of the manual, and it is far superior to all previous editions.

The DOS drivers and filters are also presented in an easy to understand manner. My wife has always avoided using filters because she didn't take the time to understand them. However, she read the instructions in this manual, and in no time had she set up CLICK/FLT and KSM/FLT. "Not difficult", she informed me.

Job Control Language (JCL) is the final tutorial. It is an extensive 44 pages of information, with each command thoroughly explained.

The LDOS & LS-DOS Reference Manual closes with a glossary of the 'computerese' terms used in the book, followed by a handy index.

The bottom line is, that if you use LDOS, LS-DOS or both, you should definitely get this manual. What a relief it is to have all the information in one place - and, written by Roy Soltoff, you know that the information is accurate.

The LDOS & LS-DOS Reference Manual is now available from Misosys, Inc, P.O. Box 239, Sterling, VA 22170, (703) 450-4181. The price is \$30.00

Next issue we'll take a look at the companion manual - the LDOS & LS-DOS BASIC Reference Manual.

HINTS & TIPS

MODEL 4 POWER SUPPLY FAULT

By Karl Mohr

Phil, a friend of mine, mentioned to me some time ago that he was having problems with his Model 4 in respect that the screen would flicker occasionally. We came to the conclusion that there was probably a poor connection somewhere between the power supply or on the video circuit board itself. As this fault appeared very randomly, we thought we would leave it until it got worse, hopefully making it easier to find.

At about the same time, another friend (Bill) starting having trouble with his Model 4. Although the problem appeared the same with the screen flickering, it acted very much different than Phil's computer did. It seemed that the problem occurred around the same time of the day, about 4PM, just when people would start preparing the evening meal. With the heavy load on the power lines, the line voltage would start dropping and as we all know, if the voltage drops low enough, it causes computer screens (T.V. screens) to shrink. This is what appeared to be happening with Bill's computer. The screen would shrink briefly, then go back to normal size, as if a heavy current draw had occurred.

Bill spoke to the Power company, but I don't know if they ever came and checked the lines in Bill's area.

Phil's Model 4 got worse with time just as we predicted. We had a look at it a few days ago to see if we could find the fault, and came up with the answer in just a few minutes. The problem was on the power supply, where the connectors plug onto the power supply pins. We found that the solder that held the 12 volt video supply pins to the foil on the back of power supply had developed a hair line crack around several of the pins. The solder would make and break its connection at random times and cause the screen to go wild. We resoldered the affected pins, in fact, we resoldered all the connection pins to ensure good solder joints. To date, Phil reports that he has had no more problems with screen jitters!

Bill brought his Model 4 over the next day to see if we could find any fault with his machine. It turned out that the problem was in the identical spot as Phil's computer. We resoldered all the pins on the power supply in Bill's computer and it's been working great ever since.

I did my own computer several months back, with the problem being the same and in the identical place. That makes three machines that have had a similar problems in a very short time. If any of the readers are experiencing problems with the screen flickering, I would suggest checking the solder at these points. It's very easy to check

and repair this problem. With the computer top off, unplug the connectors to the power supply, remove the 4 screws that hold the board in place then check the solder at these points. Use a magnifying glass if you need and be in a good light. If you see cracks in the solder around these pins, simply resolder them and your problems should be cured.

THE TRSDOS 1.3 DIRECTORY.

By Bruce McDowell

This article will describe and explain the important features of the Model III TRSDOS directory. It is quite similar in most respects to the Model I directory, except for the way in which the DOS SYSTEM files are handled.

As in the Model I, the directory is located on Track #17 (11H), and occupies the entire track. Because of the Model III's double density format, this amounts to 18, 256 byte sectors. The first sector (#0) contains the Granule Allocation Table (GAT), the diskette Master Password encode, the diskette name and any AUTO command which may be active. Sector #1 contains the Hash Index Table (HIT) and the information by which TRSDOS is able to locate the SYSTEM files. The remaining sixteen sectors of the directory track contain the directory entries for the diskettes user files, five entries per sector.

Let's take a look at Track #17 in detail:

1. Sector #0 (the GAT sector)

The first 40 bytes (00H to 27H) make up the GAT and represent Tracks 0 to 39 on the diskette. Each byte is a "bit map" of the allocated granules in the corresponding track. Bits 0 through 5 represent the 6 granules of the track and bits 6 and 7 are not used (always 0). Thus, a fully allocated track would appear as a '3F' in the GAT. In binary this would be 00111111. If a byte is '25', a binary 00100101, it means that the first, third and sixth granules are allocated.

Bytes CEH and CFH of Sector #0 are the encoded Master Password for the diskette. This is normally D38F for "PASSWORD". Bytes D0H - DFH are the diskette's name in ASCII and the creation date in MM/DD/YY format. Bytes E0H - FFH are used to store any AUTO command which may have been set up for the diskette. If byte E0H is a '0D' (carriage return), no AUTO command is in effect.

2. Sector #1 (the HIT sector)

The first 80 bytes (00H to 4FH) make up the Hash Index Table (HIT). Each non-zero byte is the hash code for an active file in the directory, and its position in the HIT

indicates in which sector and where in the sector the directory entry is located. For example, on a standard Radio Shack TRSDOS diskette, the first two bytes are 'F0' and 'F4', which are the hash codes for BASIC/CMD and CONVERT/CMD. A glance at Sector #2 will confirm that these files are the first two entries in that sector.

The Model III uses the same algorithm as the Model I for computing hash codes.

Bytes E0H - FDH are used to store the size and location of the TRSDOS SYSTEM files. There are 15 pairs of bytes and each pair represent one of the SYS files. The first byte of the pair contains the granule offset and the size of the file in granules. The second byte of the pair is the number of the track in HEX where the file starts. To see how this works, let's take a look at bytes E4H-E5H which are '4210'.

The '42' is 01000010 in binary notation. Because the byte has two pieces of information embedded in it, we'll write it as 010 00010. The '00010' is the granule count for the file, which is TWO. The '010' is the granule offset from the start of the track which is also TWO. Remember that a granule is three sectors in the Model III. The second byte of the pair is '10' which means that the file starts on Track #16 (10H).

Putting it all together, we see that the file starts on Track #16 (10H), beginning on Sector #6 (offset of TWO granules) and occupies TWO granules or six sectors. This particular file, by the way, is very similar to SYS2/SYS on the Model I. The fifteen pairs of bytes in this area represent the TRSDOS resident module and fourteen overlay modules making up the entire DOS. The sixteenth pair is presently 'FFFF' and is not in use, no doubt being reserved for a future overlay, like a spooler (just a guess). Since the user never calls the DOS modules directly, they need not have names and the information in these byte pairs is all that TRSDOS needs in order to locate them when program execution calls for them.

3. Sectors #2 through #17 (Named directory entries)

There is room for five, three line (48 byte) directory entries on each of these sixteen sectors. This gives a total capacity of 80 named files. The sixteenth line on each sector is not used, so Radio Shack placed a '(c) 1980 Tandy' on it as a filler.

Let's look at the first entry in Sector #2 to see how it is constructed. This is the entry for BASIC/CMD and it looks like this:

```
E508 5000 0042 4153 4943 2020 2043 4D44
0000 EF5C 1200 0E06 FFFF FFFF FFFF FFFF
FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
```

The first byte, 'E5', contains the file type, its visibility and protection level. In binary this is 01011110 and we'll write it as 0 1 0 1 1 110 to make it easier to break it down. Bits 0-2 are the protection level. In this case, it is 6 (EXECute only). Bit 3 is the visibility bit. A '1' means that the file is

invisible, as in this case. If bit 4 is a '1', the file is an active file and has a corresponding HIT entry. Bit 5 is usually '0', but if it should be a '1', it means that the file has a backup limitation. If bit 6 is a '1', the file is a SYSTEM file. If it is a '0', it is a USER file. Bit 7 is '0' for a Primary directory entry and is a '1' for an EXTENDED directory entry.

The second and third bytes are simply the file creation month and year in HEX, in this case '08 50', meaning 08/80. The fourth byte shows the location of the EOF byte in the file's final sector. The fifth byte is the Logical Record Length of the file (00 means 256 bytes). The sixth through the sixteenth bytes are the file name in ASCII, in this example, BASIC CMD.

Bytes 17-18 and 19-20 are the UPDATE and ACCESS password encodes, respectively. For this file, we see '0000 EF5C'. 'EF5C' is the code for NO password, so BASIC/CMD has an UPDATE password assigned but no access password.

Bytes 21-22 indicate the total number of sectors in the file in LSB-MSB format. So BASIC/CMD has 18 (12H) sectors.

The next 26 bytes are arranged in pairs and are used for the file EXTENTS. 'FFFF' indicates no further extents. The first byte is the number in HEX where the extent begins. In our example, this is '0E' meaning Track #14. The second byte of the pair contains the file's granule count and offset, encoded in exactly the same way as for the TRSDOS system modules. For BASIC/CMD, this byte is '06' which is 000 00110 in binary notation. Thus BASIC/CMD occupies 6 granules and starts at the beginning of the track (no offset).

FILEOSOPHY 101

by Allen Jacobs

To me, the MS-DOS world seems to discourage most users from programming. That's because virtually everything we need to do has already been done. This may be good for people who use computers only for a specific task, or for those hobbyists who only like to work with specific applications on their machines. However, for COMPUTER hobbyists from the "golden age of computing", (ie., the beginning -- or whenever we decide when the golden age was or will be), the inevitably sorry condition we have always dreaded is upon us. That is the time when virtually every task we will ever need to program has already been done; not really but just seems that way.

To explain, I was presented with a problem at work. A person I know had difficulty with some word processing files. He lost access to the word processor he used to create them. It was called WRITE ASSIST and I believe it was from IBM. Anyway, Word Perfect wouldn't read the files and Word Perfect was the only word processor he had. Thus, he was at a loss to recover them.

I decided to try recovery with the tools I had available. I was carrying around a copy of Q-Edit which would read

the file. I had the old version, however, that did not include a displayable ASCII character list so I could not identify the actual values of the characters I was looking at. The file appeared to consist of about 50% graphics characters. I knew that these characters were non-ASCII because the values were above 128. The problem is in determining if these characters are flagged text or strictly control characters.

I remembered that QDOS has a Wordstar filter that simply displays the high bit (7) in a reset condition. With that, I could read the characters as ASCII test, only to see that they were repeating patterns of various types, but not words. I therefore assumed that I could remove them without sacrificing any actual text. Also, I knew that the file could be edited. However, I didn't to spend hours cleaning up that file and about twenty others like it, by hand. There had to be a better way.

I was going to write a BASIC program to read each character from a file and write it to another file, only if it was 127 or below. I was going to pattern the utility after a Lewis Rosenfelder program I looked up in his book.

Instead, that night I mentioned what I was going to do at a TUG-NET meeting. A Word Perfect user told me about a utility in the program for just this problem. It's called Convert. I was sure that my problem was solved. I was surprised and pleased that such a program existed. When I tried CONVERT, it asked me the format of the file I would like it to work on. After trying one of the two reasonable file types, Convert told me that the file type was incompatible. The other file type selection converted the file instantly, without an error message and promptly wrote the destination file out to disk. It was one of the fastest file writes I have ever seen Word Perfect perform; for good reason. The file was ZERO bytes long. Nonetheless, I am grateful for TUG-NET's advice. How could they have known?

So, it was back to the "BASIC"s. Then, I remembered that I had a shareware disk from Best Bits and Bytes entitled, ASCII FILE MODIFIERS. I thought that I might as well check it out before I started writing. Well, wouldn't you know it. There was MODIFILE, a utility that was almost exactly what I had in mind. It had a couple of additional features I hadn't yet thought of; and it was written in compiled BASIC. I can only guess its lineage but it does CR to CR/LF conversions and back. Guessing what type of machine the programmer who wrote the program had, before he wrote this utility is easy. He called the conversions TRS-80 to MS-DOS. Even though the file conversion I had in mind was between MS-DOS editors, it took a program of TRS-80 lineage to do the job. Of course, it worked.

I was happy that it did; and I was happy that a TRS-80 person did it. But also, I was a little sad to find out that there wasn't anything for me to do, but watch it run....

YES, OF COURSE !

WE VERY MUCH DO TRS-80 !

MICRODEX CORPORATION SOFTWARE

CLAN-4 Mod-4 Genealogy archive & charting **\$69.95**
Quick and easy editing of family data. Print elegant graphic ancestor and descendant charts on dot-matrix and laser printers. *True Mod-4 mode*, fast 100% machine language. Includes 36-page manual. **NEW!**

XCLAN3 converts Mod-3 Clan files for Clan-4 **\$29.95**

DIRECT from CHRIS Mod-4 menu system **\$29.95**
Replaces DOS-Ready prompt. Design your own menus with an easy full-screen editor. Assign any command to any single keystroke. Up to 36 menus can instantly call each other. Auto-boot, screen blanking, more.

xT.CAD Mod-4 Computer Drafting **\$95.00**
The famous general purpose precision scaled drafting program! Surprisingly simple, yet it features CAD functions expected from expensive packages. Supports Radio Shack or MicroLabs hi-res board. Output to pen plotters. *Includes a new driver for laser printers!*

xT.CAD BILL of Materials for xT.CAD **\$45.00**
Prints alphabetized listing of parts from xT.CAD drawings. Optional quantity, cost and total calculations.

CASH Bookkeeping system for Mod-4 **\$45.00**
Easy to use, ideal for small business, professional or personal use. Journal entries are automatically distributed to user's accounts in a self-balancing ledger.

FREE User Support Included With All Programs !

MICRODEX BOOKSHELF

MOD-4 by CHRIS for TRS/LS-DOS 6.3 **\$24.95**

MOD-III by CHRIS for LDOS 5.3 **\$24.95**

MOD-III by CHRIS for TRSDOS 1.3 **\$24.95**

Beautifully designed owner's manuals completely replace obsolete Tandy and LDOS documentation. Better organized, with more examples, written in plain English, these books are a *must for every TRS-80 user*.

JCL by CHRIS Job Control Language **\$7.95**

Surprise, surprise! We've got rid of the jargon and JCL turns out to be simple, easy, useful and fun. Complete tutorial with examples and command reference section.

Z80 Tutor I Fresh look at assembly language **\$9.95**

Z80 Tutor II Programming tools, methods **\$9.95**

Z80 Tutor III File handling, BCD math, etc. **\$9.95**

Z80 Tutor X All Z80 instructions, flags **\$12.95**

Common-sense assembly tutorial & reference for novice and expert alike. Over 80 routines. No kidding!

Add S & H. Call or write MICRODEX for details
1212 N. Sawtelle Tucson AZ 85716 602/326-3502

Obituary to a TRS-80 Column (But Not This One)

by Michael W. Ecker, Ph.D.

I usually write a column on recreational and educational computing about every second or third issue for TRSTimes. This time I'd like to use this space to inform readers of a latest nail in the coffin of the long-dead TRS-80s.

As some of you know, I have been privileged for the last two years to write the only remaining TRS-80 column in a nationally published magazine. I began writing "Your TRS-80 Column" for Computer Monthly under the pseudonym of David Wade in mid 1990. The pseudonym was used in order to avoid a hassle from the Computer Shopper, whose senior editor Charlie Cooper specially called me up to "advise" not to write for Computer Monthly if I wanted to continue to write for him. That this demand stunk to high heaven I knew, but the Shopper was my cash cow and it did, after all, give me many more assignments than Computer Monthly did. (And, in fact, Computer Monthly hasn't agreed to give me one in a year now either, so my fears were not unfounded.)

It was under this backdrop of economic blackmail that David Wade was born briefly, the amalgam of my son's first name (David) and my middle name (Wade).

Only a bit later after Computer Shopper dropped me - dumped is a better word - along with two-thirds of the writers in late 1990 to early 1991, I was free to use my own name, which I have used to date in writing "Your TRS-80 Column."

Free to use my own name - just as Computer Monthly's editor wanted. Such freedom I did not need, I might add, for in the process, Computer Shopper cheated me out of \$5,000 that I have never been paid. This was for the many stories I wrote on assignment in the last half-year of 1990 but which were never used. But the egregious manner in which all the Ziff-Davis publications (e.g., Computer Shopper) treat writers is another story best told elsewhere - and besides, I might want to write for one of those SOBs again and don't want to be quoted accurately later. (So what that these people don't know how to match a pronoun and antecedent? They pay well.)

Which brings us back to the current year. As of the end of June 1992, the Classic Computer columns in Computer Monthly, including the TRS-80 column, are history. In late January 1992, Doug Kilarski, editor-in-chief, called to advise me to cut my column to about 500 words. Having written some 350 articles in my life to date, I was experi-

enced enough to expect another call in a few months to say the column was dead. In this regard I was not disappointed when I got the call in early April.

An interesting aside about the ethics of editors: I just wrote that I was not disappointed when I got the call mentioned. What was disappointing was being told the feeble line that my last column had not been received so there might be no kill fee. (Trust me: An editor who tells you that quickly that he never got your last column knows damned well he did. If he did not receive it, he would say that he would have to check.) The point of this digression is to ask what process exists that transforms every computer writer - such as Kilarski was years ago - who becomes an editor into a blood-sucking crook who will cheat a poor freelance writer out of hard-earned money.

To my good buddy and fellow computer journalist Fred Blechman, who has been responsible for getting me several writing gigs, I promise to let you know the answer to this question if any reader can tell me this.

There is a certain freedom in relative poverty: I am not writing a single column, article, or review for pay for anybody after June 1992. So, as the song goes, "When you've got nothing you've got nothing to lose."

Okay, I admit it's all sour grapes, and if Computer Shopper or Byte or PC Sources or Government Computer News or any of the other s.o.bs who have ever stopped using my writing called me back tomorrow, the following day I would be working on a story, ever the journalistic prostitute. Heaven help me, because I love this work so much I would do a lot of it free. (Come to think of it, I do do a lot of it for free.) Most writers would say something similar, I think.

Back to TRSTimes... In the course of writing the Computer Monthly TRS column, I have repeatedly been pleased to plug this magazine as the best source of TRS-80 information around. I never pretended to be a TRS-80 authority - or any other kind, for that matter - and have always seen my role as a caretaker, the guy to point new owners of old TRS-80s where to go for help.

I might add that I initially also pointed readers to Stan Slater and Computer News 80. However, that fellow went off the deep end with me and so I stopped giving him free publicity, even though I repeatedly offered to assist him in numerous ways. In that regard, I thank Chris Fara of Microdex for his kind letter and commiserate with him regarding his being "excommunicated" by dictator Slater.

If anybody would care to know, this was for Chris's criminal act (hope you all recognize the sarcasm) of daring to want to promote his own company and earn a living. Shame on you and me, Chris, for wanting to do something good for ourselves and readers!

Now, I spoke earlier of another nail in the coffin. TRSTimes, for whatever faults it may have - as do we all - is still the best source for keeping up. So, death?

Editor Lance Wolstrup has done a tremendous job of keeping TRS-80s alive, and we are all fortunate to have him. Thanks, Lance, and keep up the good work.

Still, it is possible that a rare few of you want ever more. So, should any misguided reader actually want them, I am offering all my TRS columns written for Computer Monthly, including parts cut out for space or otherwise, plus unprinted columns, and all back columns (including any you missed), for \$10 plus \$2.90 priority mail shipping. This buys you 70 very full laser-printed pages.

And, if I may, I would like to use Lance's willingness to print my remarks to excerpt some of what you won't see printed in Computer Monthly but which I had submitted. After all, it kills me to do the work and not have this printed.

Graphics 90

I finally got a working copy of Graphics 90. This is one of the best graphic programs ever for the low-resolution TRS-80s, now being made available once again. Incredibly, the new publisher (TRSTimes) is offering it at cost for subscribers to TRSTimes.

Graphics 90 creates stunning animation sequences. Available for several different TRS-80 DOSes, its ABASIC adds commands to ordinary BASIC.

In a special offer, the whole package is available for \$19.75, which includes shipping. However, for just \$10 more, you can also get a one-year subscription to TRSTimes (5721 Topanga Canyon Blvd. #4/ Woodland Hills, CA 91367; phone: (818) 716-7154). Those with current TRSTimes subs already may get Graphics 90 for just \$9.75. (My advice in my Computer Monthly column was to spring for the whole \$29.75 package, but if you're reading this, you're probably a TRSTimes subscriber already, so I'd just be preaching to the already converted.)

World's Smallest Word Processor

In an earlier column I had issued some programming challenges. TRSTimes Editor Lance Wolstrup sent this BASIC program entry to turn your printer (connected to Model 1 or 3) into a typewriter:

```
1 LPRINT INKEY$;: GOTO 1
```

You won't be able to make corrections. For Model 4, change GOTO 1 to RUN.

Tape Spreadsheet

I've received several requests now for information about a TRS-80 tape-based spreadsheet. Can anybody advise of such software not requiring a disk system? (The only one I knew about was Nova Calc in 80 Micro around 1984 to 1986, but I found it slow as molasses.)

and Recreations Stop Calling '#' a Pound Sign!

Mr. Dave Park of Glendale, MO wrote to answer a question I had asked in my (former) TRS-80 column in Computer Monthly a year or so ago:

What is the name of the "#" symbol?

Most people unhesitatingly call this the pound sign and are familiar with it from keyboards (shift 3 on most PCs) and telephones. But I had heard another name long ago and asked readers for help. Dave filled me in:

"Dear Mike,

Maybe I can be of help. In return, you can sign up with my new organization ('Stop Calling it a Pound Sign'). A pound sign is that funny cursive capital L with a crossbar that the British use to signify a pound sterling.

The word you are seeking to name the symbol of two vertical and two horizontal lines criss-crossing is...

OCTOTHORP.

At the time I was introduced to the word (in connection with telephone touch-dialing), it was explained to me that 'octothorp' comes from two Greek words meaning 'eight' and 'point'; trouble is, I can't find the word in my English dictionary... I do know that Western Electric had issued an internal press release about the symbol to coin the term, and then, a week or two later, issued another release saying, in effect, to disregard the previous release... There's a good story behind this, I'm sure... Regardless (of whether coinage is a hoax or legitimate), I like 'octothorp'; it's much more genteel than the 'pound sign' nomenclature being passed off as genuine currency."

Thanks, Dave... Okay, how about a little recreation before closing?

Digital Teaser

Yoshiyuki Kotani wrote me from Japan for a timely teaser. He wrote:

"To: Dr. Michael W. Ecker -- Dear Mike,

1) We can satisfy the formula $A \times B \times AB = 1992$ for this year. What are the unknown digits A and B?

2) What year equal to the value of $A \times B \times AB$ for some digits A and B will come next?"

Nice teaser - which, in fact, you can do even without a computer. Readers?

The Power of Powers

Dr. Mietek Szyszkowicz sent in this "little school problem" to think about:

"Without using calculators, computers, logarithms, etc., determine the largest number from among these: 2^{777} , 3^{666} , 4^{555} , 5^{444} , 6^{333} , 7^{222} ."

Okay, got it. Readers?... And now, on to solutions and the mail...

Free Model I

Ms. Jo Dittert (P.O. Box 99/ Bellville, TX 77418) is offering a free TRS-80 Model 1 to anybody who promises to repair it. There are absolutely no other strings attached, and it's yours to keep. She just wants it not to go to waste or storage. Any hacker care to take it off her hands?

Solutions

Self-Listing Programs

Brian Dozier of McPherson, KS sent a self-listing program in response to my challenge. Such a program produces the same output regardless of whether you issue the command LIST or RUN from BASIC. His used data statements among the 32 lines, and if he had sent a disk I would have shared his solution, which is not unlike one I featured in my "Recreational Computing" column in Popular Computing or Creative Computing about eight years ago.

Fortunately, there is a much more concise solution:

```
1 A$="1 A$=2 PRINT LEFT$(A$,5);CHR$(34);A$:
PRINT RIGHT$(A$,51)
2 PRINT LEFT$(A$,5);CHR$(34);A$:
PRINT RIGHT$(A$,51)
```

If memory serves, Norlin Rober sent this to me six years ago. But a more recent one is even tighter, a mere 86 characters on one line. It is the last of five different solutions sent last year by Dr. Fred Galvin, Professor of Mathematics at the University of Kansas. Fred's solution is:

```
1 X$=" "+CHR$(34):PRINT MID$(X$,35)+X$+X$'1
X$=" "+CHR$(34):PRINT MID$(X$,35)+X$+X$'1 X$=" "
```

Note that you should be able to count exactly 86 characters and/or spaces (which may not show up clearly in the listing in this magazine). Do not issue a carriage return (don't press ENTER) even if listing shows more than one line.

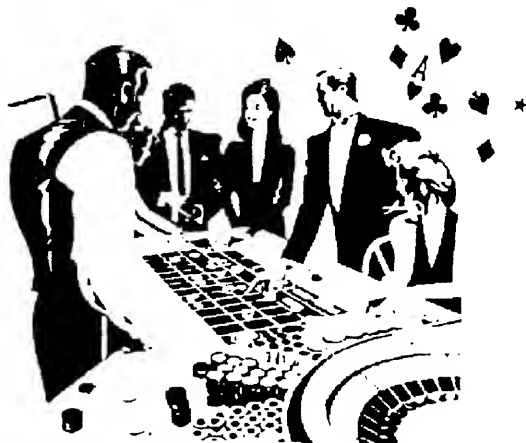
Mathematical Recreations Collections

If you would like to get started with a self-booting disk in a menu-driven format filled with logic games, send \$10 plus \$2 shipping/handling to Recreational Computing/ 909 Violet Terrace/Clarks Summit, PA 18411. Games include Maxit, Teaser, MasterMind, the L Game, and a whole bunch more whose names I've momentarily forgotten. There are also a few other bonus programs not on the menu system. Specify Model 3, 4, or 4P.

And, while I'm making this shameless commercial pitch, let me add one more: I have a modest supply of about 50 to 75 different CLOAD magazine tapes for sale at \$6.50 each. Each contains great games, education, utilities, and much more. Buy five of these unique items and I'll include a sixth for free, but hurry, because I have only one or two of any one. Write me at the same address above.

That's it for now. Time to play a game of Sea Dragon in tribute to my Model 3, my first computer. Until next time, columnist for hire thanks you for reading.

Dr. Michael W. Ecker, holder of a Ph.D. in mathematics, is a Pennsylvania State University mathematics professor as well as a computer writer-reviewer and columnist with 350 publication credits. The author of two books, Mike is also Editor/Publisher of Recreational & Educational Computing (or REC), from which parts of this column were drawn. Now in its seventh year (1992), REC is available for \$27 per year in USA or \$36 outside North America. Readers of this magazine are invited to try a trial subscription of three sample issues (\$10 in USA or \$13 outside North America).



TRSTimes Special PD Collection

Model I & III

PD01*

APD/ASM, APD/DOC, APD1/CMD, APD3/CMD - automatic pdrive recognizer.

COMPDIR/BAS, COMPDIR/DOC - compare 2 directories.

DISKINDX/CMD, DISKINDX/DOC - index newdos disks.

NAMEIT/BAS, NAMEIT2/BAS - customize newdos startup message.

PDRIVE/BAS - pdrive analyzer.

REROUTE/ASM, REROUTE/CMD - reroute printer output to file.

SUPERLST/CMD, SUPERLST/DOC - print basic programs uncompressed.

TRSDIR/ASM, TRSDIR/CMD, TRSDIR/DOC - get trsdos directories from newdos.

* note all files on this disk work on NEWDOS/80 v.2. only.

PD02

LHELP/TXT, LTERM/CMD, LTERM/DOC - good communications program.

PILOT/BAS, PILOT/INS - the pilot programming language.

REPLACER/BAS, REPLACER/DOC - replace variables in basic programs.

SCHEDULE/BAS - a monthly planner.

PD03

CALC/CMD - make calculations directly from dos.

CHECKER/BAS - checks ascii files.

DIALND/CMD - stern dialer for newdos.

DISKCAT/BAS, DISKCAT/DOC - disk cataloger for dosplus.

LOADADDR/BAS - finds /cmd file load address.

MODEM/CMD, MODEM/DOC

NEWLIST/CMD, NEWLIST/DOC - basic program lister for dosplus.

NOTE/CMD - make notes to display on the crt.

ORCONV/CMD - convert orch-90 files.

STERM/CMD, STERM/DOC - great communications program by Larry Payne.

XMOD1200/CMD, XMOD300/CMD, XMODEM/DOC - xmodem file transfer.

PD04

BASICOMP/BAS, BASICOMP/DOC - a basic compiler.
CLRDIR/CMD - zero unused directory entries - for mod I/III multidos.

FDCIII/BAS - test your floppy disk controller.

FREEMAP/CMD - free-space map for multidos.

KBMOD/ASM, KBMOD/CMD - move cursor anywhere on screen - mod 4 in III mode.

MEMDISK4/CMD - memory disk drive for 128k mod 4 in 3 mode.

OTHELLO/BAS - the classic board game.

SOUND134/BAS - sound routines for mod I/III & 4.

PD05

ALARM/BAS, ALARM/DOC - protect your model I.

BINCLOCK/ASM, BINCLOCK/CMD,

BINCLOCK/DOC - a fun binary clock.

CONCEN/BAS - the game of concentration.

DIRSLOT/ASM, DIRSLOT/DOC - direct any file to selected directory slot - newdos.

DR/ASM, DR/CMD, DR/DOC - super route command for newdos.

TRSPATCH/BAS - patches for trsdos 1.3.

VCLIST/BAS, VCLIST/DOC - visicalc file listing utility.

PD06

BOOT/BAS - create custom boot message on newdos.

D/CMD, D/DOC - sorted directory listing for newdos.

DISTANCE/BAS - figure distance between two cities.

FLIGHTDC/BAS, FLIGHTSM/BAS - flight simulator.

LINEBAS, LINE/DOC - line drawing program - demo.

MINIBBS2/BAS, MINIBBS2/DOC, MINIBBS2/JCL,

MINIHOST/ASM, MINIHOST/CMD - complete bulletin board system.

SUNRISE/BAS - figure sunrise/sunset anywhere in the world.

XMODEM30/CMD - modem file transfer utility.

PD07

BLANK/ASM, BLANK/CMD, BLANK1/CMD, DEMO - screen blanker for newdos.

CLAWDOS/CMD, CLAWDOS/DOC, CLAWDOS/SRC - dos shell for trsdos 1.3.

CLONEI/CMD, CLONEI/DOC, CLONEI/SRC - copy utility for model III - trsdos 1.3 or newdos.

EMPIRE/BAS - a game of conquest.

FNDEMO/BAS - demo of basic functions.

GRAPHPRO/BAS, GRAPHPRO/DOC - program makes print statements from the graphics you draw on the screen.

SUPCALC2/BAS, SUPCALC2/DOC - super calculator.

TRACE/ASM, TRACE/CMD, TRACE/DOC - trace utility for newdos.

PD08

ASCFMT/ASM, ASCFMT/CMD - read ascii files - view on screen or send to printer.

CONVBIN/ASM, CONVBIN/CMD - convert binary numbers.

DIS/CMD, DIS/DOC, DIS/SRC - sorted directory for newdos/80.

FORM/CMD, FORM/DOC, FORM/SRC - special disk formatter for Model I/III.

MARQ/ASM, MARQ/CMD - generate a marquee around the screen.

NEWBLD/ASM, NEWBLD/CMD, NEWBLD/DOC - a better way to write /jcl files.

PHONETXT/BAS - find names in phone numbers.

PD09

EASTER/BAS, EASTER/DOC - find easter in any given year.

FILMGR48/BAS, FILMGR48/DOC, FILMAGR48/FIX, FM48/JCL, MENU/DAT, BONDS/E85, BONDS/RPG, PLATES/E85, PLATES/RPG, WIDGET/CRF, SPEED/CMD, README/DAT - database for Model III newdos/80.

MORSE/ASM, MORSE/CMD - generate morse code from ascii text.

PLANNER/BAS, PLANNER/DOC - financial planner.

PD10

CMDJDEMO/BAS, CMDJDEMO/DOC - learn how to use cmd"J" - newdos.

COPYCATD/ASM, COPYCATD/CMD, COPYCATD/DOC - unprotect copycat - and move it to a /cmd file.

DATECOMP/ASM, DATECOMP/CMD - compress the date format in your programs.

DBLPREC/BAS, DBLPREC/DOC - double precision demo.

SCRPR1/BAS, SCRPR1/CMD, SCRPR1/DOC, SCRPR1/CMD - screen printer for Model I/III.

SZAP80/ASM, SZAP80/CMD, SZAP80/DOC - patch superzap to work in 80 column mode - newdos/80.

VARLST/ASM, VARLST/CMD, VARLST/DOC - list variables to the screen.

WRDSRCH/BAS, WRDSRCH/DOC - word searches.

PD11

BASLIST/ASM, BASLIST/CMD, BASLIST/DOC - a 'pretty-printer' utility for newdos.

CAT/CMD, CAT/DOC, CAT/SRC - a utility that categorizes the disk space.

DISKCAT/CMD - disk cataloger for newdos and dosplus 3.4.

HEXCAL/CMD - hex calculator for newdos.

OYSTER/BAS - make the system your oyster.

ROM/ASM, ROM/CMD, ROM/DOC - upper bank memory transfer utility - Model 4 in III mode - newdos.

TRACK/CMD - a complete video tracking program.

PD12

COMPARE/ASM, COMPARE/CMD - compare any two files on a disk.

FONTDVR/ASM, FONTDVR/CMD, FONTDVR/DOC - driver for fontwrtr..

FONTWRTR/ASM, FONTWRTR/CMD, FONTWRTR/DOC - create character sets for epson printers

GOSTMENU/ASM, GOSTMENU/CMD, GOSTMENU/DOC - interesting menu program for Model I/III newdos.

PD13

EDTFMT/ASM, EDTFMT/CMD - format edtasm text to printer.

HEAPSORT/BAS - heapsort demonstration program.

MAZE/ASM, MAZE/BAS, MAZE/CIM, MAZE/DOC, MAZEGEN/BAS - create interesting mazes.

SHELLSORT/BAS - shellsort demonstration program.

UTI/CMD, UTI/DOC - compare of list files or directories.

VFU/CMD, VFU/DOC - versatile file utility for newdos or dosplus 3.4.

PD14

CLSLDOS/JCL - patch to implement cls command for Idos 5.1.3.

D/CMD - nice menu program for newdos - fast.

KEYMAC/ASM, KEYMAC/CMD, KEYMAC/DOC - keyboard utility for Model I/III.

LDOSLOGO/CMD - personalize Idos 5.1.x boot logo.

MAPLDOS/CMD - map files on double density Idos disks.

NDCAT/ASM, NDCAT/CMD - sorted directory for newdos.

NEWMAP/BAS, NEWMAP/DOC - map utility for newdos.

PRINTER/DOC, PRINTER3/CMD, PRINTER3/SRC, PRINTER4/CMD, PRINTER4/SRC - set up epson printer from newdos.

UNMACRO/ASM, UNMACRO/CMD - remove keymac from memory.

PD15

CLEANER3/ASM, CLEANER3/CMD, CLEANER3/DOC - a great utility for cleaning disk drives.

DIFKIT1/BAS, DIFKIT1/DOC - visicalc utility.

MX80LBL/BAS, MX80LBL/DOC - diskette label maker for epson mx80 printers.

PDRAW/CMD, PDRAW/DOC - clever drawing program.

PRINTGRF/BAS, PRINTGRF/DOC - speed up printer graphics - epson.

RESTORE/CMD, RESTORE/DOC - recover deleted files - newdos.

TRSTimes Special PD Collection

1 to 10 volumes	\$3.00 per volume + \$2.00 postage
11 to 15 volumes	\$2.75 per volume + \$3.00 postage

TRSTimes magazine - dept. special PD
5721 Topanga Canyon Blvd. Suite #4
Woodland Hills, CA 91367

SALUTE TO THE FLAG

Model 4 - LS-DOS 6.3.x - BASIC

by Lance Wolstrup



What could be more appropriate than to pay tribute to the flag in this, our July issue. While 'old Glory' is honored on a daily basis at the Wolstrup house, the flag we will discuss in this article, however, is of a somewhat different nature. We will look at the 'heart and soul' of LS-DOS 6.3.x: the *flags table*.

The author(s) of TRSDOS/LS-DOS 6 chose to use 26 consecutive low-memory locations (from 6AH to 83H) to store pertinent information about the DOS. These memory locations are called the *flags table*, and are commonly referred to as the A-flag through the Z-flag.

Not all 26 locations are used; to the best of my knowledge the B, G, H, Q, X, Y, and Z flags have been left open for possible (but unlikely) future expansion of the DOS. The U-flag is designated the 'user-flag'; that is, you, the user/programmer, can use this flag to store one byte of information important to your program. The remainder of the alphabet-named locations all store crucial system information, some which can be played with, others which are better left for DOS.

For example, the S-flag (SYSTEM flag) can be altered by the user to disable/enable the BREAK key, as well as several other things. On the other hand, the O-flag (OPREG\$ mask) is probably better left alone, unless you REALLY know what you are doing. Otherwise, messing with that one almost guarantees a freeze-up or reboot.

Many of the flags are 'bit-maps'; that is, each bit in the particular byte control certain inner workings of DOS. This, of course, makes the flag-table values difficult to decipher even for experienced users.

To make the flags-table known to all, and help me remember, I wrote FLAGS/BAS. It is a simple, fairly straight-forward Basic program that displays the entire flag table, the address of each flag, as well as the value found at each address, in decimal, hexadecimal, and binary. Further, pressing the name of a flag (A-Z), displays all the information about the particular flag that I have been able to gather.

Keep in mind that the program does not change any flags-table values; rather, it simply displays information. It is intended as a learning tool or as a programmer's refer-

ence utility. There are no machine language routines, though I should probably had used one to speed up the flags-table display, which is just a smidgen on the slow side because it is converting the value found at each flag address from decimal to binary. Oh well, here's to straight Basic!

FLAGS/BAS

```
0 'FLAGS/BAS
1 'for Model 4 - LS-DOS 6.3.x
2 'copyright (c) 1992 by Lance Wolstrup
3 'all rights reserved
10 DEFINT A-Z
11 SW = 80
12 B$(1) = "Flag Adr":B$(2) = "Dec Hex Binary"
15 PRINT CHR$(15);:GOTO 100
20 H = 0:GOTO 23
21 H = INT((SW-LEN(A$))/2):GOTO 23
22 H = SW-LEN(A$)
23 PRINT@SW*V + H,A$;:RETURN
50 PRINT HEX$(Z);" ";
51 PRINT USING"###";PEEK(Z);
PRINT" ";A$ = HEX$(PEEK(Z));
IF LEN(A$) < 2 THEN A$ = "0" + A$
52 PRINT A$;" ";
53 N = PEEK(Z):GOSUB 90
54 RETURN
60 V = 5:H = 3:A$ = B$(1):GOSUB 23:
H = 15:A$ = B$(2):GOSUB 23:RETURN
61 H = 43:A$ = B$(1):GOSUB 23:
H = 55:A$ = B$(2):GOSUB 23:RETURN
70 V = 10:FOR X = 0 TO 7:A$ = "bit" + STR$(X) + " - ";
GOSUB 20:V = V + 1:NEXT V = 10:H = 8:A$ = "":
GOSUB 23:RETURN
90 BN = N:Y = 256:FOR XX = 1 TO 8:Y = Y/2:
IF BN/Y = > 1 THEN BN = BN-Y:PRINT"1";
ELSE PRINT"0";
91 IF XX = 4 THEN PRINT" ";
92 NEXT:RETURN
100 CLS:V = 0:A$ = "TRSTimes presents:":GOSUB 20:
A$ = "F L A G S":GOSUB 21:
A$ = "(c) 1992 Lance Wolstrup":GOSUB 22:V = 1:
A$ = "display LS-DOS 6.3.1 FLAGS table information":
GOSUB 21:V = 2:A$ = STRING$(SW,131):GOSUB 20
105 GOSUB 60:GOSUB 61
110 Z = &H6A:V = 6:H = 4:FOR X = 0 TO 25:
IF X > 12 THEN V = X-7:H = 44
120 A$ = CHR$(X + 65) + "":GOSUB 23:GOSUB 50
160 Z = Z + 1:V = V + 1:NEXT A$ = STRING$(SW,131):
```



```

GOSUB 23
200 V = 22:H = 5:
A$ = "Press letter of flag to examine - or press CTRL-Q
to quit":GOSUB 23
210 H = 63:A$ = CHR$(14):GOSUB 23
220 I$ = INKEY$:IF I$ = "" THEN 220
230 I = ASC(I$):IF I = 17 THEN CLS:END
240 IF I < 65 THEN 220
250 IF I > 90 THEN IF I < 97 OR I > 122 THEN 220
260 IF I > 90 THEN I = I-32
270 PRINT CHR$(15);
300 V = 3:A$ = CHR$(31):GOSUB 20:GOSUB 60:
V = 6:H = 4:A$ = CHR$(I) + " ":GOSUB 23:
Z = I + 41:GOSUB 50:V = 8
310 ON I-64 GOTO 1000,1100,1200,1300,1400,1500,
1100,1100,1800,1100,2000,2100,2200,2300,2400,2500,
1100,2700,2800,2900,3000,3100,3200,1100,1100,1100
1000 A$ = "Start cylinder for Allocation search":
GOSUB 20
1010 V = 10:
A$ = "Allocation search starts on cylinder" +
STR$(PEEK(Z)):GOSUB 20
1020 GOTO 4000
1100 A$ = "This flag is not defined":GOSUB 20:
GOTO 4000
1200 A$ = "Condition flag":GOSUB 20
1210 GOSUB 70
1220 A$ = "cannot change HIGH$ via SVC-100":
GOSUB 23
1221 V = V + 1:A$ = "@CMDR in execution":GOSUB 23
1222 V = V + 1:A$ = "@KEYIN request from SYS1":
GOSUB 23
1223 V = V + 1:
A$ = "system request for drivers, filters, DCTs":
GOSUB 23
1224 V = V + 1:
A$ = "@CMDR to execute only LIB commands":
GOSUB 23
1225 V = V + 1:A$ = "SYSGEN inhibit bit":GOSUB 23
1226 V = V + 1:A$ = "@ERROR inhibit bit":GOSUB 23
1227 V = V + 1:A$ = "@ERROR to use user (DE) buffer":
GOSUB 23
1230 GOTO 4000
1300 A$ = "Device flag":GOSUB 20
1310 GOSUB 70
1320 A$ = "SPOOL is active":GOSUB 23
1321 V = V + 1:A$ = "TYPE ahead is active":GOSUB 23
1322 V = V + 1:A$ = "VERIFY is on":GOSUB 23
1323 V = V + 1:A$ = "SMOOTH active":GOSUB 23
1324 V = V + 1:A$ = "memDISK active":GOSUB 23
1325 V = V + 1:A$ = "FORMS active":GOSUB 23
1326 V = V + 1:A$ = "KSM active":GOSUB 23
1327 V = V + 1:A$ = "accept GRAPHICS in screen print":
GOSUB 23
1330 GOTO 4000
1400 A$ = "Entry code to be passed to SYS 13":
GOSUB 20

```

```

1410 V = V + 2:A$ = "use only bits 4,5, and 6 to indicate
user entry code to be passe to SYS13.":GOSUB 20:
V = V + 1:A$ = "SYS13 will be executed from SYS1 if this
byte is non-zero, bit 4,5, and 6":GOSUB 20
1411 V = V + 1:A$ = "will be merged into the SYS13
(1000 1111b) overlay request":GOSUB 20
1420 GOTO 4000
1500 A$ = "Port FE mask":GOSUB 20
1510 GOTO 4000
1800 A$ = "International flag":GOSUB 20
1810 GOSUB 70
1820 A$ = "FRENCH":GOSUB 23
1821 V = V + 1:A$ = "GERMAN":GOSUB 23
1822 V = V + 1:A$ = "SWISS":GOSUB 23
1826 V = V + 4:A$ = "special DMP mode ON/OFF":
GOSUB 23
1827 V = V + 1:A$ = "'7' bit mode ON/OFF":GOSUB 23
1830 GOTO 4000
2000 A$ = "Keyboard flag":GOSUB 20
2010 GOSUB 70
2020 A$ = "BREAK latch":GOSUB 23
2021 V = V + 1:A$ = "PAUSE latch":GOSUB 23
2022 V = V + 1:A$ = "ENTER latch":GOSUB 23
2023 V = V + 1:A$ = "reserved":GOSUB 23
2024 V = V + 1:GOSUB 23
2025 V = V + 1:A$ = "CAPs lock":GOSUB 23
2026 V = V + 1:A$ = "reserved":GOSUB 23
2027 V = V + 1:A$ = "character in TYPE ahead":
GOSUB 23
2030 GOTO 4000
2100 A$ = "L(S)-DOS feature inhibit flag":GOSUB 20
2110 GOSUB 70
2120 A$ = "inhibit step rate question in FORMAT":
GOSUB 23
2124 V = V + 4:A$ = "inhibit 8" + CHR$(34) + " query in
FLOPPY/DCT":GOSUB 23
2125 V = V + 1:
A$ = "inhibit # of sides question in FORMAT":GOSUB 23
2126 V = V + 1:A$ = "reserved for IM 2 hardware":
GOSUB 23
2127 V = V + 1:GOSUB 23
2130 GOTO 4000
2200 A$ = "MODOUT$ mask - image of port 0ECH":
GOSUB 23
2210 GOSUB 70
2221 V = V + 1:A$ = "cassette motor on/of":GOSUB 23
2222 V = V + 1:A$ = "mode select (0 = 80/64,
1 = 40/32)":GOSUB 23
2223 V = V + 1:A$ = "enable alternate character set":
GOSUB 23
2224 V = V + 1:A$ = "enable external I/O":GOSUB 23
2225 V = V + 1:
A$ = "video wait states (0 = disable, 1 = enable)":
GOSUB 23
2226 V = V + 1:A$ = "clock speed (1 = 4 mhz, 0 = 2 mhz)":
GOSUB 23
2230 GOTO 4000
2300 A$ = "Network flag":GOSUB 20

```



```

2310 GOSUB 70
2320 A$="allow setting of of file open bit in DIR":
GOSUB 23
2321 V=V+1:A$="reserved":GOSUB 23
2322 V=V+1:GOSUB 23
2323 V=V+1:GOSUB 23
2324 V=V+1:GOSUB 23
2325 V=V+1:GOSUB 23
2326 V=V+1:A$="set if in Task Processor":GOSUB 23
2327 V=V+1:A$="file password protection disabled":
GOSUB 23
2330 GOTO 4000
2400 A$="OPREG$ memory management image port":
GOSUB 20
2410 GOSUB 70
2420 A$="SEL0 - select map overlay bit 0":GOSUB 23
2421 V=V+1:A$="SEL1 - select map overlay bit 1":
GOSUB 23
2422 V=V+1:A$="80/64 (1 = 80x24, 0 = 64x16)":
GOSUB 23
2423 V=V+1:A$="inverse video":GOSUB 23
2424 V=V+1:A$="MEMBIT0 - memory map bit 0":
GOSUB 23
2425 V=V+1:A$="MBIT1 - memory map bit 1":
GOSUB 23
2426 V=V+1:A$="FXUPMEM - fix upper memory":
GOSUB 23
2427 V=V+1:A$="PAGE - page 1K video RAM (set for
80x24)":GOSUB 23
2430 GOTO 4000
2500 A$="Printer flag":GOSUB 20
2510 GOSUB 70
2520 A$="reserved":GOSUB 23
2521 V=V+1:GOSUB 23
2522 V=V+1:GOSUB 23
2523 V=V+1:GOSUB 23
2524 V=V+1:GOSUB 23
2525 V=V+1:GOSUB 23
2526 V=V+1:GOSUB 23
2527 V=V+1:A$="printer spooler is paused":
GOSUB 23
2530 GOTO 4000
2700 A$="FDC Retry count":GOSUB 20
2710 V=V+2:A$="Retry" + STR$(PEEK(Z)) + " times":
GOSUB 20
2720 GOTO 4000
2800 A$="System flag":GOSUB 20
2810 GOSUB 70
2820 A$="inhibit file open bit":GOSUB 23
2821 V=V+1:
A$="set to 1 if bit 2 is set and EXEC file opened":
GOSUB 23
2822 V=V+1:
A$="set by @RUN to permit load of EXEC file":
GOSUB 23
2823 V=V+1:A$="SYSTEM (FAST)":GOSUB 23
2824 V=V+1:A$="BREAK key disabled":GOSUB 23
2825 V=V+1:A$="JCL active":GOSUB 23

```

```

2826 V=V+1:A$="force extended error messages":
GOSUB 23
2827 V=V+1:A$="DEBUG to be turned off after load":
GOSUB 23
2830 GOTO 4000
2900 A$="Type flag":GOSUB 20
2910 V=V+2:A$=" 2 = Model II":GOSUB 20
2911 V=V+1:A$=" 4 = Model 4":GOSUB 20
2912 V=V+1:A$=" 5 = Model 4P":GOSUB 20
2913 V=V+1:A$="12 = Model 12":GOSUB 20
2914 V=V+1:A$="16 = Model 16":GOSUB 20
2915 V=V+2:A$="This machine is a Model ":
GOSUB 20
2920 Z=PEEK(Z):IF Z=2 THEN PRINT"II"
ELSE IF Z=4 THEN PRINT"4"
ELSE IF Z=5 THEN PRINT"4P"
ELSE IF Z=12 THEN PRINT"12"
ELSE PRINT"16"
2930 GOTO 4000
3000 A$="User flag":GOSUB 20
3010 V=V+2:
A$="This flag may be assigned at the user's discretion":
GOSUB 20
3020 GOTO 4000
3100 A$="Video flag":GOSUB 20
3110 GOSUB 70
3120 A$="set blink":GOSUB 23
3121 V=V+1:H=17:A$="rate":GOSUB 23
3122 V=V+1:H=21:A$="(1 = fastest,":GOSUB 23
3123 V=V+1:H=32:A$="7 = slowest)":GOSUB 23
3124 V=V+1:H=8:A$="display CLOCK":GOSUB 23
3125 V=V+1:A$="cursor blink toggle bit":GOSUB 23
3126 V=V+1:A$="inhibit blinking cursor (user)":
GOSUB 23
3127 V=V+1:A$="inhibit blinking cursor (system)":
GOSUB 23
3130 GOTO 4000
3200 A$="WRINT$ - interrupt mask register":GOSUB 20
3210 GOSUB 70
3220 A$="enable 1500 baud rising edge":GOSUB 23
3221 V=V+1:A$="enable 1500 baud falling edge":
GOSUB 23
3222 V=V+1:A$="enable real time clock":GOSUB 23
3223 V=V+1:A$="enable I/O bus interrupts":
GOSUB 23
3224 V=V+1:A$="enable RS-232 transmit interrupts":
GOSUB 23
3225 V=V+1:
A$="enable RS-232 receive data interrupts":
GOSUB 23
3226 V=V+1:A$="enable RS-232 error interrupt":
GOSUB 23
4000 V=21:A$=STRING$(SW,131):GOSUB 20:
V=22:A$="Press any key to continue " + CHR$(14):
GOSUB 21
4010 I$=INKEY$:IF I$="" THEN 4010
ELSE PRINT CHR$(15);:V=3:A$=CHR$(31):
GOSUB 20:GOTO 105

```


PROGRAMMING TIDBITS

Copyright 1992 by Chris Fara, Microdex

Basic Days of the Week



Browsing through Dr.Ecker's "TRS-80 Column" in "Computer Monthly", in the December 1990 issue I noticed a routine to calculate the day of the week for any date. A handy thing for many applications. So I was about to jot it down in my notebook, when Dr.Ecker's comment about the routine's math caught my attention: "a bit beyond me", he wrote. Well, as a math professor, at that time hiding under a pen-name "David Wade", he was perhaps exaggerating a bit for effect. But I had to admit that the math was at least beyond me. Unnecessarily convoluted, it did a great job of obscuring the simple logic of the calendar. A habitual itch to clarify and simplify kept me awake until I cooked up a more straightforward scheme.

```
10 INPUT "Year, Month, Day "; y,m,d
20 x = y + (m)
30 y = y-1 + x\4 - x\100 + x\400
40 m = (m-1 + (m2))\2 + (m-(m7))\2
50 d = (y+m+d) MOD 7
```

The first idea here is that starting from any arbitrary date, each next year shifts the weekdays by one day (there are 52 weeks plus one day in a normal year). This is expressed in line 30 by Y-1. Each leap year shifts the weekdays by one more day, except in January and February. The intermediate variable X in line 20 accounts for this "except": a BASIC comparison such as (M) yields -1 if true, 0 if false. Thus in line 30 the three "integer divisions" say: add one day for each leap year (X\4) except in century years (X\100), but do count century years divisible by 400. The result Y is the sum of shifts caused by elapsed years.

If all months had 28 days then weekdays would be the same in each month. This not being the case, line 40 calculates shifts caused by months. The first expression adds 2 shifts for each month except February. The second part adds one extra day every other month, corrected for the back-to-back 31-day months of July and August.

In line 50 add up those shifts plus the day of the month, and calculate the remainder of division by 7 (modulo 7). The result is a number 0-6. To synchronize it with the "real" calendar look up any known date and add a constant to Y+M+D. As it turns out, the constant is zero for our

current "Gregorian" calendar. Hence conveniently D=0 for Sunday, D=1 Monday, and so on.

In those BASICs that don't sport the backslash "integer division", use the FIX function. For example instead of X\4 write FIX(X/4). Similarly replace the MODulo operator with expressions such as...

```
50 d = y+m+d: d = d - FIX(d/7)*7
```

To spell out weekday names add something like this...

```
60 w$ = "SunMonTueWedThuFriSat"
70 PRINT MID$(w$, d*3+1, 3)
```

Prior to 1583 Europe and its colonies used the "Julian" calendar established in 46 BC by Julius Caesar for the Roman empire. All century years, not just those divisible by 400, used to be leap years. As a result the calendar was running too slow. To catch up, the Gregorian system deleted 10 days from people's life (October 5-14, 1582), which caused some wild street riots! But no riot is needed to modify our routine for the old Julian dates. Simply shorten line 30...

```
30 y = y-1 + x\4
```

and in line 50 add a constant 5...

```
50 d = (y+m+d+5) MOD 7
```

In the English-speaking world, including US territories, the mess was even worse. The Gregorian system was not adopted until 1752 when 11 days were "lost" (September 3-13). Not only that: before the switch New Year's day used to be March 25! Care to tweak our algorithm to calculate that "Old Style" birthday of your grand-grand-grand...?

Speaking about "julian" dates. Some BASICs, for example Mod-III, have a command to convert a date to a consecutive number of the day in a year. They call it "julian" date, but it has nothing to do with the old Roman calendar, nor is it quite like the Julian dating system known in astronomy. Astronomers count consecutive days starting from January 1, 4713 BC, a theoretical date when the solar and lunar cycles would have been in synch with the ancient Roman tax collection cycle (...and you thought IRS was our modern American invention). By now those day numbers are truly astronomical, close to 2,450,000. The term "julian" was coined in 1582 by the author of this dating system, the astronomer Scaligeri, in memory of his father Julius.

Model 4 Public Domain Disks

M4GOODIES#1: day/cmd, day/txt, gomuku/cmd, llife/cmd, llife/doc, writer/cmd, writer/doc, writer/hlp, yahtzee/bas

M4GOODIES#2: arc4/cmd, arc4/doc, cia/bas, etimer/cmd, index/cmd, index/dat, mail/bas, mail/txt, trscat/cmd, trscat/txt, util4/cmd, xt4/cmd, xt4/dat, xt4hlp/dat

M4GOODIES#3: convbase/bas, dates/bas, dctdsp/cmd, dmu/cmd, dmu/doc, dskcat5/cmd, dskcat5/doc, editor/cmd, editor/doc, fedit/cmd, fkey/asm, fkey/cmd, fkey/doc, hangman/cmd, m/cmd, m/src, membrane/bas, miniop2/cmd, miniop2/src, move/cmd, move/doc, othello4/bas, scroll4/cmd, scroll4/src, setdate6/cmd, setdate6/doc, setdate6/fix, spaceadv/bas, taxman/bas, utilbill/bas, utilbill/doc

M4GOODIES#4: WORD WIZARD disk #1 of 3
anima/bas, archi/bas, autos/bas, battuere/bas, captus/bas, convert/bas, curro/bas, dico/bas, ducere/bas, eulogos/bas, facere/bas, fluere/bas, gradi/bas, jacere/bas, kata/bas, male/bas, metron/bas, naus/bas, startup/bas, startup/jcl, stig/bas, tangere/bas, wordmenu/bas

M4GOODIES#5: WORD WIZARD disk #2 of 3
cognos/bas, frangere/bas, juris/bas, medius/bas, mittere/bas, monos/bas, numbers/bas, orare/bas, pandemos/bas, para/bas, pathos/bas, pendere/bas, philanth/bas, phongrap/bas, polynom/bas, prefix1/bas, prefix2/bas, premere/bas, sal/bas, startup/bas, startup/jcl, statuere/bas, wordmenu/bas

M4GOODIES#6: WORD WIZARD disk 3 of 3
bible/bas, french1/bas, french2/bas, french3/bas, italian/bas, latphras/bas, lit1/bas, lit2/bas, myths/bas, places/bas, plicare/bas, spanish/bas, stagnare/bas, stare/bas, startup/bas, startup/jcl, synpath/bas, televid/bas, tenere/bas, vaco/bas, valere/bas, vox/bas, wordmenu/bas

M4GOODIES#7: calendar/cmd, castladv/bas, civilwar/bas, crimeadv/bas, dctdsp/cmd, ed6/cmd, ed6/doc, edittext/bas, fedit/cmd, mail/bas, mail/txt, scramble/bas, states/bas, textpro/cmd, time4/bas, wizard/bas, wizard/doc, worldcap/bas

M4GOODIES#8: books/bas, books/doc, dmu/cmd, dmu/doc, hamcalc/bas, hamhelp/bas, network/bas, network/doc, pirate/bas, pirate/doc, vmap/bas, vmap/doc, vmap2/bas, vmap2/doc, zork1/doc, zork2/doc, zork3/doc
M4GOODIES#9: ft/cmd, ft/doc, pterm/cmd, pterm/doc, r/cmd, r/doc, scrconv/bas, scrconv/doc, video4/asm, video4/cmd

M4GOODIES#10: checker/cmd, crossref/cmd, crossref/doc, ddir/cmd, diskcat/cmd, diskcat/doc, division/bas, division/doc, getput/bas, getput/doc, host/cmd, hv/bas, maszap4/cmd, maszap4/doc, park/cmd, profile4/doc, protect/bas, protect/doc, rename/bas, replace/bas, re-

store/bas, rm/bas, scrndump/bas, scrndump/doc, super/hlp, vers/cmd

M4GOODIES#11: benchmrk/bas, bigcal/bas, bigcal/doc, birthday/bas, dearc4/cmd, dezip2/cmd, dname/cmd, docufil/bas, docufil/doc, docufil/mrg, escape/bas, mem4/cmd, million/bas, nomad/bas, password/bas, password/dat, password/doc, password/jcl, roman/bas, sixtymin/bas, startrek/bas, trekinst/bas

M4GOODIES#12: awari/bas, buying/bas, crasher/bas, curvfit2/bas, gradebk/bas, mortcost/bas, mortcost/doc, prlnt/bas, print/doc, reiman/bas, square/bas, starlane/bas, staybus/bas, sunrise/bas, synonym/bas, timezon1/bas, timezon2/bas, travel/bas, vmap2/bas, vmap2/doc, weekday/bas

M4GOODIES#13: calndr1/bas, calndr2/bas, calndr3/bas, formltrs/bas, invloan/bas, limerick/bas, martian/bas, mission/bas, moneymkt/bas, munchmth/bas, numbrfun/bas, smith/bas, smith/doc, star2000/bas, starfind/bas, starfind/dat, starfind/doc, starfind/jcl, states/bas, wallst/bas

M4GOODIES#14: alphahex/bas, bowlchng/bas, bowicrea/bas, bowldetl/bas, bowlfinl/bas, bowling/doc, bowlmenu/bas, bowlprnt/bas, bowlrcap/bas, bowirecd/bas, bowirecp/bas, bowlschd/bas, bowlscor/bas, bowisort/bas, buscheck/bas, calculat/bas, chekform/bas, deprec/bas, futrdate/bas, membrain/bas, minimath/bas, normalz/bas, numconv/bas, pcbdest/bas, pcbdest/doc, pcform/bas, pcpm/bas, pcpm/doc, pcpm/jcl, utscan/bas, yagibeam/bas, zeller/bas

M4GOODIES#15: laughs/bas, laughs/dat, laughs/doc, laughs1/dat, laughs2/dat, laughs3/dat, laughs4/dat, laughs5/dat, laughs6/dat, laughs7/dat, laughs8/dat, laughs9/dat, laughs10/dat, laughs11/dat, laughs12/dat, laughs13/dat, laughs14/dat, laughs15/dat

M4GOODIES#16: trivia/bas, trivia/doc, trivia1/dat, trivia2/dat, trivia3/dat, trivia4/dat

M4GOODIES#17: acrs/bas, amorloan/bas, clockmod/bas, compound/bas, dcform/bas, decide/bas, easyword/bas, editno/bas, epslabel/bas, esckey/bas, expect/bas, funct1/bas, funct2/bas, gasform/bas, hexprint/bas, hexsay/bas, lostgold/bas, mathfunc/bas, mpgcalc/bas, neclabel/bas, nicelist/bas, nonlin/bas, nonlin/rem, payback/bas, peekprnt/bas, percent/bas, prntcall/bas, proverbs/bas, randseed/bas, savings/bas, speech/bas, tasklist/bas, tempconv/bas, weightfm/bas

**Each disk is \$5.00 (U.S.)
or get any 3 disks for \$12.00 (U.S.)
please specify the exact disks wanted.**

**TRSTimes PD-DISKS
5721 Topanga Canyon Blvd., Suite 4
Woodland Hills, CA. 91367**

HARD DRIVES FOR SALE

Genuine Radio Shack Drive Boxes with Controller, Power Supply, and Cables.
Formatted for TRS 6.3, installation JCL included.
Hardware write protect operational.
Documentation and new copy of MISOSYS RSHARD5/6 included.
90 day warranty.

5 Meg \$175

10 Meg \$225

15 Meg \$275

35 Meg \$445

Shipping cost add to all prices



Roy T. Beck
2153 Cedarhurst Dr.
Los Angeles, CA 90027
(213) 664-5059



TIRED OF SLOPPY DISK LABELS? TIRED OF NOT KNOWING WHAT'S ON YOUR DISK? YOU NEED 'DL'

'DL' will automatically read your TRSDOS6/LDOS compatible disk
and then print a neat label, listing the visible files (maximum 16).
You may use the 'change' feature to select (or reject) the filenames to print.
You may even change the diskname and diskdate.
'DL' is written in 100% Z-80 machine code for efficiency and speed.

Don't be without it - order your copy today.

'DL' is available for TRS-80 Model 4/4P/4D
using TRSDOS 6.2/LS-DOS 6.3.0 & 6.3.1.
with either an Epson compatible or DMP series printer.

'DL' for Model 4 only \$9.95

TRSTimes magazine - Dept. 'DL'
5721 Topanga Canyon Blvd. #4
Woodland Hills, CA 91367

ATTENTION TRSDOS 1.3. USERS!

ANNOUNCING "SYSTEM 1.5.", THE MOST COMPREHENSIVE 1.3. UPGRADE EVER OFFERED!

MORE SPEED!! MORE POWER!! MORE PUNCH!!

While maintaining 100% compatibility to TRSDOS 1.3., this DOS upgrade advances TRSDOS 1.3. into the 90's!
SYSTEM 1.5. supports 16k-32k bank data storage and 4MGHZ clock speed (4/4P/4D).

DOUBLE SIDED DRIVES ARE NOW 100% UTILIZED! (all models).

CONFIG=Y/N
TIME=Y/N
BLINK=Y/N
LINE='XX'
ALIVE=Y/N
TRON=Y/N
TYPE=B/H/Y/N
SLOW
CPY (parm,parm)
SYSRES=Y/N
SPOOL=H/B.SIZE
SPOOL=N
SPOOL=RESET
SPOOL=CLOSE
FILTER *PR.IGLF
FILTER *PR.FILTER
FILTER *PR.FIND
FILTER *PR.LINES
FILTER *PR.TMARG
FILTER *PR.PAGE
FILTER *PR.TOF
FILTER *KI.ECHO
ATTRIB:d.PASSWORD

CREATES CONFIG BOOT UP FILE
TIME BOOT UP PROMPT ON or OFF
SET CURSOR BOOT UP DEFAULT
SET *PR LINES BOOT UP DEFAULT
GRAPHIC MONITOR ON or OFF
ADD an IMPROVED TRON
HIGH/BANK TYPE AHEAD ON or OFF
2 MGHZ SPEED (MODEL III'S)
COPY/LIST/CAT LDOS TYPE DISKS
DISABLE/ENABLE SYSRES OPTION
SPOOL is HIGH or BANK MEMORY
TEMPORARILY DISABLE SPOOLER
RESET (NIL) SPOOL BUFFER
CLOSES SPOOL DISK FILE
IGNORES 'EXTRA' LINE FEEDS
ADDS 256 BYTE PRINTER FILTER
TRANSLATE PRINTER BYTE TO CHNG
DEFINE NUMBER LINES PER PAGE
ADDS TOP MARGIN to PRINTOUTS
NUMBER PAGES, SET PAGE NUMBER
MOVES PAPER TO TOP OF FORM
ECHO KEYS to the PRINTER
CHANGE MASTER PASSWORD

DATE=Y/N
CURSOR='XX'
CAPS=Y/N
WP=d.Y/N (WP)
TRACE=Y/N
MEMORY=Y/N
FAST
BASIC2
SYSRES=H/B/'XX'
MACRO
SPOOL=D.SIZE='XX'
SPOOL=Y
SPOOL=OPEN
FILTER *PR.ADLF=Y/N
FILTER *PR.HARD=Y/N
FILTER *PR.ORIG
FILTER *PR.RESET
FILTER *PR.WIDTH
FILTER *PR.BMARG
FILTER *PR.ROUTE
FILTER *PR.NEWPG
FILTER *KI.MACRO
DEVICE

DATE BOOT UP PROMPT ON or OFF
DEFINE BOOT UP CURSOR CHAR
SET KEY CAPS BOOT UP DEFAULT
WRITE PROTECT ANY or ALL DRIVES
TURN SP MONITOR ON or OFF
BASIC FREE MEMORY DISPLAY MONITOR
4 MGHZ SPEED (MODEL 4'S)
ENTER ROM BASIC (NON-DISK)
MOVE/SYS OVERLAY(s) TO H/BANK MEM
DEFINE ANY KEY TO MACRO
LINK MEM SPOOLING TO DISK FILE
REACTIVATE DISABLED SPOOLER
OPENS, REACTIVATES DISK SPOOLING
ADD LINE FEEDS BEFORE PRINTING 0DH
SEND 0CH to PRINTER (FASTEST TOF)
TRANSLATE PRINTER BYTE TO CHNG
RESET PRINTER FILTER TABLE
DEFINE PRINTER LINE WIDTH
ADDS BOTTOM MARGIN to PRINTOUT
SETS PRINTER ROUTING ON or OFF
SET DCB LINE COUNT TO 1
TURN MACRO KEYS ON or OFF
DISPLAYS CURRENT CONFIG INFO

All parms above are installed using the new LIBRARY command SYSTEM (parm,parm). Other new LIB options include DBSIDE (enables double sided drive by treating the "other side" as a new independent drive, drives 0-7 supported) and SWAP (swap drive code table #s). Dump (CONFIG) all current high and/or bank memory data/routines and other current config to a disk data file. If your type ahead is active, you can (optional) store text in the type buffer, which is saved. During a boot, the config file is loaded back into high/bank memory and interrupts are recognized. After executing any active auto command, any stored type ahead data will be output. FANTASTIC! Convert your QWERTY keyboard to a DVORAK! Route printer output to the screen or your RS-232. Macro any key, even F1, F2 or F3. Load *01-*15 overlay(s) into high/bank memory for a memory only DOS! Enter data faster with the 256 byte type ahead option. Run 4MGHZ error free as clock, disk I/O routines are properly corrected! Spool printing to high/bank memory. Link spooling to disk (spooling updates DCB upon entering storage). Install up to 4 different debugging monitors. Print MS-DOS text files, ignoring those unwanted line feeds. Copy, Lprint, List or CATalog DOSPLUS, LS-DOS, LDOS or TRSDOS 6.x.x. files and disks. Add top/bottom margins and/or page numbers to your hard copy. Rename/Redate disks. Use special printer codes eg: LPRINT CHR\$(3); toggles printer output to the ROUTE device. Special keyboard codes add even more versatility. This upgrade improves date file stamping MM/DD/YY instead of just MM/YY. Adds optional verify on/off formatting, enables users to examine *01-*15, DIR, and BOOT sectors using DEBUG, and corrects all known TRSDOS 1.3. DOS errors. Upgrade includes LIBDVR, a /CMD driver that enables LIBRARY commands, such as DIR, COPY, DEBUG, FREE, PURGE, or even small /CMD programs to be used within a running Basic program, without variable or data loss.

By special arrangement with GRL Software,
SYSTEM 1.5. is now distributed exclusively by TRSTimes magazine.

ORDER YOUR COPY TODAY!

Send \$39.95 (U.S. funds) to:

TRSTimes - SYSTEM 1.5.
5721 Topanga Canyon Blvd., Suite 4
Woodland Hills, CA. 91367

***** DR. PATCH *****

BRAND-NEW UTILITY FOR TRS-80 MODEL 4 AND LS-DOS 6.3.1

A 'MUST HAVE' FOR ALL LS-DOS 6.3.1 OWNERS.

**DR. PATCH MODIFIES LS-DOS 6.3.1 TO DO THINGS THAT WERE NEVER BEFORE POSSIBLE.
COMPLETELY SELF-CONTAINED - MENU-DRIVEN FOR MAXIMUM USER CONVENIENCE.
FAST & SAFE - EACH MODIFICATION IS EASILY REVERSED TO NORMAL DOS OPERATION.**

DISABLE PASSWORD CHECK IN FORMAT/CMD
FORMAT DOUBLE-SIDED AS DEFAULT
FORMAT 80 TRACKS AS DEFAULT
DISABLE VERIFY AFTER FORMAT
CHANGE 'DIR' TO 'D'
CHANGE 'CAT' TO 'C'
VIEW DIR/CAT WITH (I) PARAMETER AS DEFAULT
VIEW DIR/CAT WITH (S,I) PARAMETERS AS DEFAULT
CHANGE 'REMOVE' TO 'DEL'
CHANGE 'RENAME' TO 'REN'
CHANGE 'MEMORY' TO 'MEM'
CHANGE 'DEVICE' TO 'DEV'
DISABLE THE BOOT 'DATE' PROMPT
DISABLE THE BOOT 'TIME' PROMPT
DISABLE FILE PASSWORD PROTECTION
ENABLE EXTENDED ERROR MESSAGES

DISABLE PASSWORD CHECK IN BACKUP/CMD
BACKUP WITH (I) PARAMETER AS DEFAULT
BACKUP WITH VERIFY DISABLED
DISABLE BACKUP 'LIMIT' PROTECTION
DISABLE PASSWORD CHECK IN PURGE
PURGE WITH (I) PARAMETER AS DEFAULT
PURGE WITH (S,I) PARAMETERS AS DEFAULT
PURGE WITH (Q=N) PARAMETER AS DEFAULT
IMPLEMENT THE DOS 'KILL' COMMAND
CHANGE DOS PROMPT TO CUSTOM PROMPT
TURN 'AUTO BREAK DISABLE' OFF
TURN 'SYSGEN' MESSAGE OFF
BOOT WITH NON-BLINKING CURSOR
BOOT WITH CUSTOM CURSOR
BOOT WITH CLOCK ON
BOOT WITH FAST KEY-REPEAT

**DR. PATCH IS THE ONLY PROGRAM OF ITS TYPE EVER WRITTEN FOR THE TRS-80
MODEL 4 AND LS-DOS 6.3.1. IT IS DISTRIBUTED EXCLUSIVELY BY TRSTIMES' MAGAZINE
ON A STANDARD LS-DOS 6.3.1 DATA DISKETTE, ALONG WITH WRITTEN DOCUMENTATION.**

***** DR. PATCH *** \$14.95**

SHIPPING & HANDLING: U.S & CANADA - NONE
ELSEWHERE - ADD \$4.00
(U.S CURRENCY ONLY, PLEASE)

**TRSTimes magazine - dept. DP
5721 Topanga Canyon Blvd. #4
Woodland Hills, CA 91367**

DON'T LET YOUR LS-DOS 6.3.1 BE WITHOUT IT!

GRAPHICS-90

for Model III TRSDOS 1.3/1.5 or LDOS

The definitive graphics program that Radio Shack **SHOULD** have released is now, through the courtesy of author Larry Payne and the hard work of Gary Shanafelt, available from TRSTimes. Our agreement is to sell this wonderful package at **COST**. Neither Larry Payne, Gary Shanafelt nor TRSTimes will make a dime on this venture. The price charged will cover the cost of reproducing the manual, disks, and mailing only. The only one profiting is **YOU!!**

GRAPHICS-90 \$6.75

Shipping: U.S. & Canada: add \$3.00

Europe: add \$6.25 air mail or \$4.25 surface mail

Asia, Australia & New Zealand: add \$7.25 air mail or \$4.25 surface mail

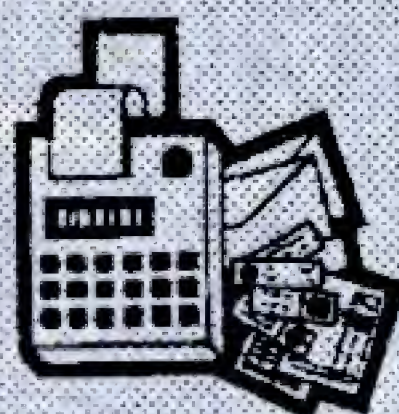
Please specify TRSDOS 1.3 or LDOS format

TRSTimes magazine - dept. GR90

5721 Topanga Canyon Blvd. #4

Woodland Hills, CA 91367

RECREATIONAL & EDUCATIONAL COMPUTING



REC is the only publication devoted to the playful interaction of computers and 'mathemagic' - from digital delights to strange attractors, from special number classes to computer graphics and fractals. Edited and published by computer columnist and math professor Dr. Michael W. Ecker, REC features programs, challenges, puzzles, program teasers, art, editorial, humor, and much more, all laser printed. REC supports many computer brands as it has done since inception Jan. 1986. Back issues are available.

To subscribe for one year of 8 issues, send \$27 US or \$36 outside North America to REC, Att: Dr. M. Ecker, 909 Violet Terrace, Clarks Summit, PA 18411, USA or send \$10 (\$13 non-US) for 3 sample issues, creditable.

TRSTimes on DISK #9

Issue #9 of TRSTimes on DISK is now available, featuring the programs from the Jan/Feb, Mar/Apr and May/Jun 1992 issues:

TRSTimes on DISK is reasonably priced:

U.S. & Canada: \$5.00 (U.S.)
Other countries: \$7.00 (U.S.)

Send check or money order to:

TRSTimes on DISK
5721 Topanga Canyon Blvd. #4
Woodland Hills, CA 91367

TRSTimes on DISK #1, 2, 3, 4, 5, 6, 7 & 8
are still available at the above prices